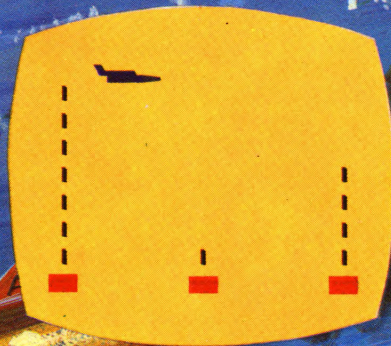


Speedy Computer

# Battaglie

con il computer



GRUPPO EDITORIALE JACKSON

per ZX81, ZX Spectrum,  
BBC, TRS-80,  
Apple, Vic  
e C 64









# Battaglie


con il computer



**Daniel Isaaman  
e Jenny Tyler**



## Indice

- 
- |   |  |
|---|--|
| 2 A proposito di questo libro             | 30 Missile                                   |
| 4 Missile Robot                           | 30 Missile versione TRS80                    |
| 6 Il messaggio vitale                     | 31 Missile versione BBC                      |
| 8 Duello                                  | 32 Missile versione ZX<br>Spectrum           |
| 10 Battaglia nel deserto                  | 33 Missile versione VIC20                    |
| 12 Battaglia al castello del<br>traditore | 34 Missile versione Apple                    |
| 14 I robot invasori                       | 35 Missile versione ZX81                     |
| 16 Arma segreta                           | 36 Aggiunte ai programmi                     |
| 18 Fuga                                   | 38 Scrivete voi stessi i<br>vostri programmi |
| 20 Caccia al pirata                       | 40 Riassunto del Basic                       |
| 22 Bombardiere supersonico                | 47 Risposte                                  |
| 24 Iceberg                                |  |
| 26 Il muro del mago                       |  |

Illustrato da  
Rex Archer, Jim Bamber, Tony Baskeyfield, Martin Newton, Graham Round.

Progetto di  
Graham Round e Roger Priddy

Traduzione: Roberto Giovannini





## A proposito di questo libro

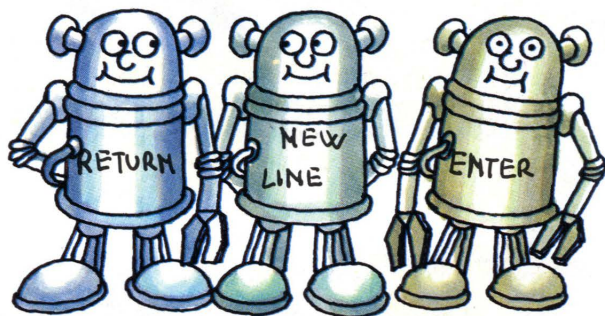
Questo libro contiene semplici programmi di giochi per personal computer. Sono stati scritti per essere usati su ZX81, ZX Spectrum, BBC, VIC 20, TRS-80, Pet ed Apple e molti di essi sono abbastanza corti da poter stare nell'unico K di memoria dello ZX81. La maggior parte dei micro usa il linguaggio BASIC, ma ognuno ha le proprie varianti o il proprio dialetto. In questo libro la versione principale di ogni programma gira sullo VIC 20 e le linee che richiedono modifiche per gli altri computer sono distinte da simboli e riportate più sotto nelle altre versioni. Il fatto che i programmi siano scritti per parecchi micro differenti implica che non fanno pieno uso delle caratteristiche di ciascuno. Potreste provare voi stessi a trovare il modo di rendere i programmi più corti ed adatti per il vostro personal. Per ogni gioco sono proposte idee per apportare cambiamenti ed aggiunte ai programmi e verso la fine del libro troverete cenni utili per scrivere voi stessi dei giochi. Nel libro è anche contenuta una tabella di conversione, per aiutarvi ad adattare al vostro personal programmi che potete trovare su riviste od altri libri, ed inoltre un riassunto dei termini BASIC usati in questo libro.

### Inserimento dei programmi da tastiera

Le linee che richiedono cambiamenti per computers diversi dallo ZX81 sono contraddistinte da questi simboli:

- ▲ ZX81
- \* BBC e Acorn Electron
- TRS-80
- Apple
- S ZX Spectrum

Ogni volta che vedrete il simbolo corrispondente al micro che state usando, guardate a fondo pagina in corrispondenza di quel numero di linea e dello stesso simbolo e battere quella linea invece di quella del programma principale.

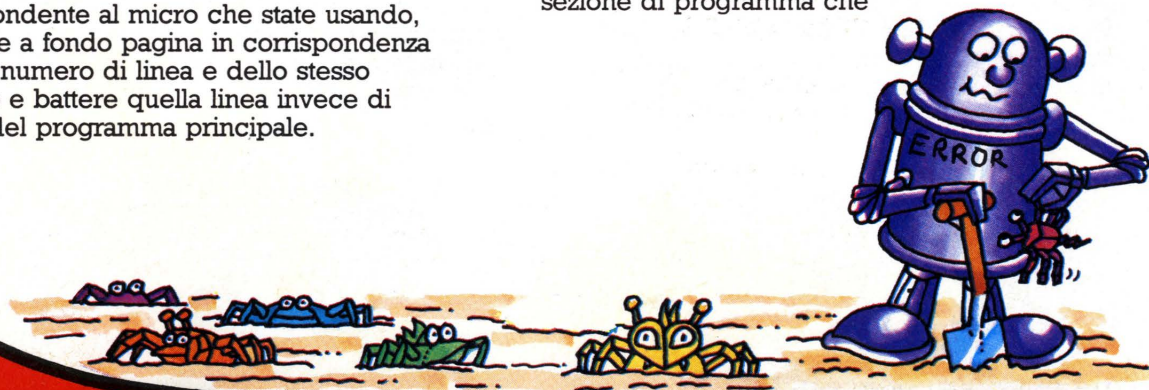


### Punti da ricordare

- 1 Battete le linee esattamente come sono scritte, compresi i segni di interpunzione e gli spazi.
- 2 Premete RETURN, NEWLINE o ENTER al termine di ogni linea di programma.
- 3 Controllate ogni linea man mano che la terminate.
- 4 Assicuratevi di non aver saltato linee e di non averne confuse. Un pezzo di carta o un righello possono servire a tenere il segno durante la copiatura.
- 5 Fate attenzione ai simboli ed assicuratevi di aver usato la linea giusta per il vostro computer.
- 6 Se usate uno ZX81 o uno ZX Spectrum, ricordate di non battere le istruzioni lettera per lettera ma di usare invece i tasti speciali per ogni istruzione. Potreste trovare più comodo farvi dettare il programma mentre lo battete. Non dimenticate di spiegare al vostro aiutante che vi deve leggere ogni virgola, punto, parentesi e spazio e differenziare la lettera "O" dallo zero.

### Ricerca degli errori

Quando avete battuto il programma, controllate il vostro manuale per vedere come farlo comparire sullo schermo. (Solitamente si scrive LIST seguito dai numeri di linea della sezione di programma che



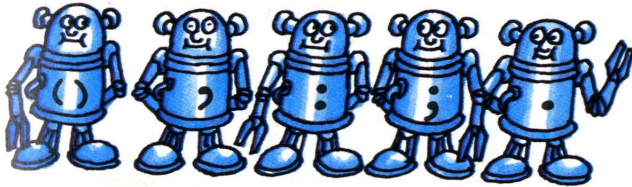


voLETE vedere). Controllate che abbiate battuto ogni cosa correttamente. È facile fare errori, perciò aspettatevi di trovarne! Usate il vostro manuale per scoprire come cambiare il programma una volta battuto. Se avete dubbi, potete sempre ribattere la linea che non vi soddisfa. Ogni computer permette di sostituire una linea esistente con una nuova avente lo stesso numero.



Ecco ora un elenco dei piú comuni errori:

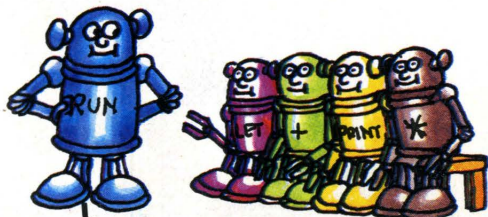
- 1 Linea mancante
- 2 Linea con numero errato
- 3 L'inizio di una linea si è unito al termine della precedente.



- 4 Parentesi, virgole, due punti, punti e virgole, punti o spazi mancanti, specialmente se le linee sono lunghe e complesse. Fate attenzione in particolare alle doppie parentesi.
- 5 Linea non adatta al vostro computer.
- 6 Confusione fra lettera "O" e zero.
- 7 Numero errato, ad es. con troppi zeri.

## I giochi

Per iniziare un gioco dovrete battere RUN; in alcuni giochi tutto accade molto velocemente, dunque leggete bene le istruzioni per sapere prima cosa dovrete fare. È probabile che a questo punto il gioco abbia ancora degli errori nel

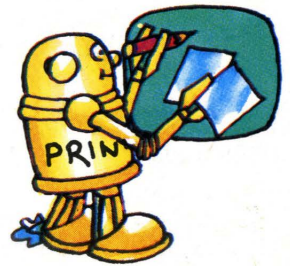


programma, cosa che gli impedirà di funzionare correttamente o addirittura totalmente. A volte il vostro computer vi darà un codice di errore che troverete spiegato nel vostro manuale. Questo potrà aiutarvi a trovare l'errore, sebbene non sempre ciò accadrà. Listate di nuovo il programma e controllatelo attentamente con il testo. Al termine del gioco, il computer dirà qualcosa del tipo BREAK IN LINE 200. Per giocare ancora, dovete battere RUN di nuovo.

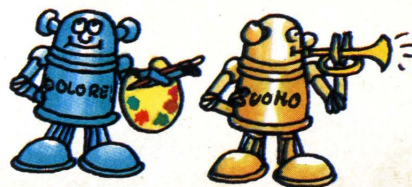
## Esperimenti con i giochi

In tutto il libro troverete suggerimenti per modifiche o aggiunte ai programmi, ma non abbiate paura se volete sperimentare qualcosa di vostro. Non danneggerete il computer e potrete comunque tornare all'originale se i cambiamenti non funzionano. Probabilmente vi troverete a voler modificare la velocità di qualche gioco\*, specialmente dopo aver giocato un certo numero di volte. Troverete sulle pagine di ciascun programma quali linee dovrete cambiare.

Ogni volta che incontrate l'istruzione PRINT, potete cambiare il messaggio fra virgolette che la segue come volete. Inoltre, a meno che non abbiate uno ZX81 con un solo K di memoria, potete aggiungere messaggi extra.



Battete un numero di linea (ad es. 105 se volete aggiungere un messaggio fra le linee 100 e 110), poi battete PRINT, quindi il vostro messaggio fra virgolette. Se il vostro computer è in grado di produrre suoni e colori, potrete usare il vostro manuale per scoprire come fare per aggiungerli ai programmi di questo libro ed avere giochi ancora piú eccitanti.



\* Ved. pag. 37 (per i possessori di BBC e Spectrum).



# Missile Robot

Siamo nell'anno 2585 e la popolazione terrestre è nel bel mezzo di una battaglia contro i Robot. Un letale missile-robot è appena atterrato e dipende da voi scoprire il codice segreto che ne sblocca il meccanismo di disinnescamento! Se fallirete l'intero Quartier Generale delle forze Terrestri esploderà. Il vostro computer sa qual'è la lettera in codice. Battete una lettera ed esso vi dirà se la lettera segreta è prima o dopo nell'alfabeto. Avete quattro tentativi per scoprire la lettera esatta prima che il missile esploda.







## Come lavora il programma

```

10 PRINT"MISSILE ROBOT"
20 PRINT
30 PRINT"SCRIVI LA LETTERA ESATTA"
40 PRINT"(A-Z) DEL CODICE CHE"
50 PRINT"DISATTIVA IL MISSILE"
60 PRINT"HAI 4 POSSIBILITA'"
70 PRINT

```

Queste righe stampano il titolo e le istruzioni.

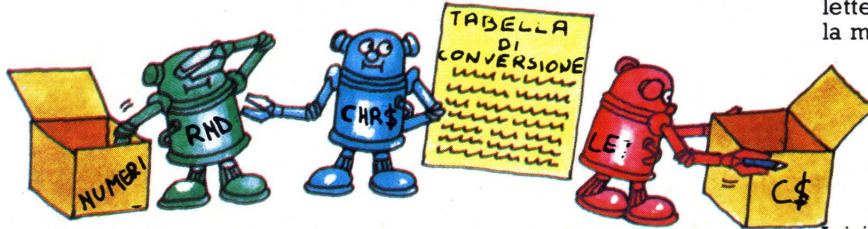


Questa sceglie un numero fra 0 e 26, lo converte in una lettera (il codice segreto!) e la memorizza in C\$.

```

▲ ■ S 80 LET C$=CHR$(64+INT(RND(1)*26+1))

```



```

90 FOR G=1 TO 4
100 INPUT G$
110 IF G$=C$ THEN GOTO 210

```

Inizia un ciclo per ripetere 4 volte le linee 100-140.

Mette il vostro tentativo in G\$.

Controlla se la vostra lettera è uguale al codice segreto in C\$. Se è così il computer salta alla linea 210.

```

120 IF G$<C$ THEN PRINT "PIU' AVANTI";
130 IF G$>C$ THEN PRINT "PIU' INDIETRO";
140 PRINT " DI ";G$

```

Queste righe controllano se la lettera segreta è, nell'alfabeto, prima o dopo la vostra e stampano un messaggio appropriato.

```

150 NEXT G

```

Fine del ciclo. Ritorna all'inizio per il tentativo seguente.

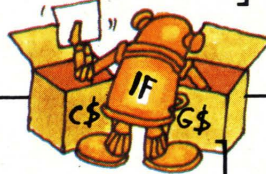
```

160 PRINT
170 PRINT"BOOOOOOOOOM..."
180 PRINT"L'HAI FATTO ESPLODERE!"
190 PRINT"IL CODICE ESATTO ERA: ";C$
200 STOP
210 PRINT"TICK...FZZZZ...CLICK..."
220 PRINT"CE L'HAI FATTA!"
230 STOP

```

Stampa nel caso di tentativi falliti.

Stampa nel caso di successo.



**Il listato sopra riportato gira su VIC 20. Per altri computer introdurre le modifiche sotto.**

```

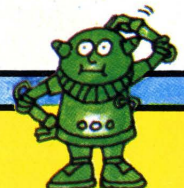
▲ 80 LET C$=CHR$(37+INT(RND*26+1))
■ 80 LET C$=CHR$(64+INT(RND(0)*26)+1)
S 80 LET C$=CHR$(64+INT(RND*26+1))

```

## Aggiunte al programma

Potete fare in modo che il computer stampi un messaggio extra se indovinate proprio all'ultimo tentativo. Cambiate la linea 220 aggiungendovi un punto e virgola in questo modo:  
 220 PRINT "CE L'HAI FATTA";  
 ed aggiungete una nuova linea 230:  
 230 IF G=5 THEN PRINT "(APPENA IN TEMPO)"

## Rompicapo



**Come si può fare in modo che il programma vi dia più o meno possibilità di indovinare la lettera segreta?**





## Il messaggio vitale

Siete un operatore di comunicazioni laser. Avete il compito di intercettare i messaggi dei robot e di riferirli al Comando del Quartier Generale. Si attende un messaggio vitale in codice. Se lo riferirete correttamente, l'attacco dei robot sarà sventato.

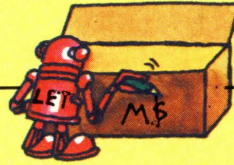
Questo gioco mette alla prova la vostra abilità di ricordare un gruppo di lettere visto solo per breve tempo. Il computer vi chiederà un livello di difficoltà da 4 a 10. Una volta data questa risposta, appariranno alcune lettere sullo schermo in alto a sinistra e poi scompariranno rapidamente. Memorizzatele e battetele sul computer: scoprirete se avete buona memoria.



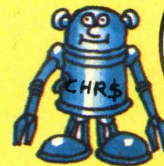


## Come lavora il programma

- \*■●▲ s 10 PRINT CHR\$(147) \_\_\_\_\_ Pulisce lo schermo prima dell'inizio del gioco.  
 20 PRINT "MESSAGGIO VITALE"  
 30 PRINT  
 40 PRINT "DIFFICOLTA' ? (4-10)" \_\_\_\_\_ } Richiede un livello di difficoltà, che viene messo in D.  
 50 INPUT D  
 60 IF D<4 OR D>10 THEN GOTO 40 \_\_\_\_\_ Controlla che il numero sia compreso fra 4 e 10.
- 70 LET M\$="" \_\_\_\_\_ Questa riga predispone una stringa vuota ("nulla" nel gergo dei computer) chiamata M\$ in cui il computer memorizzerà il messaggio segreto.
- 80 FOR I=1TOD  
 ▲■ s 90 LET M\$=M\$+CHR\$(INT(RND(1)\*26+65)) \_\_\_\_\_ Ogni volta sceglie una lettera e la aggiunge alla stringa già presente in M\$.  
 100 NEXT I
- ▲ s 110 PRINT CHR\$(147) \_\_\_\_\_ } Pulisce lo schermo e stampa il messaggio.  
 120 PRINT "TRASMETTI QUESTO MESSAGGIO:"  
 130 PRINT  
 140 PRINT M\$
- ▲ s ⊠ 150 FOR I=1 TO D\*180 \_\_\_\_\_ } Il messaggio rimane visibile mentre il computer esegue un ciclo, senza fare nulla, per un numero di volte che dipende da D.  
 160 NEXT I
- ▲ s 170 PRINT CHR\$(147) \_\_\_\_\_ Pulisce lo schermo al termine del ciclo.
- 180 INPUT N\$ \_\_\_\_\_ Mette la vostra versione del messaggio in N\$.
- 190 IF N\$=M\$ THEN GOTO 240. \_\_\_\_\_ Confronta se il vostro messaggio è uguale a quello contenuto in M\$ e, se è così salta alla linea 240.
- 200 PRINT "HAI SBAGLIATO"  
 210 PRINT "AVRESTI DOVUTO TRASMETTERE:" \_\_\_\_\_ } Queste righe vengono stampate se avete sbagliato, visualizzando il messaggio corretto.  
 220 PRINT M\$  
 230 GOTO 260  
 240 PRINT "MESSAGGIO CORRETTO"  
 250 PRINT "LA GUERRA E' TERMINATA"  
 260 STOP



Notate che lo ZX81 usa un codice differente per i caratteri della tastiera. Tutti gli altri computer usano il codice ASCII.



### Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.

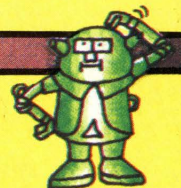
- 10,110,170 HOME
- ⊠■▲ s 10,110,170 CLS
- ▲ 90 LET M\$=M\$+CHR\$(INT(RND\*26+38))
- 90 LET M\$=M\$+CHR\$(INT(RND(0)\*26)+65))
- s 90 LET M\$=M\$+CHR\$(INT(RND\*26)+65))
- ▲ s 150 FOR I=1 TO D\*8
- ⊠ 150 FOR I=1 TO D\*400

### Come rendere il gioco più difficile

Potete cambiare il programma in modo che il messaggio segreto comprenda anche numeri e segni di interpunzione. Dovrete cambiare la linea 90 in questo modo:

- zx 90 LET M\$=M\$+CHR\$(INT(RND\*43+21))
- ⊠● 90 LET M\$=M\$+CHR\$(INT(RND(1)\*43+48))
- 90 LET M\$=M\$+CHR\$(INT(RND(0)\*43+48))
- s 90 LET M\$=M\$+CHR\$(INT(RND\*43+48))

### Rompicapo



Come si può fare in modo che il messaggio resti visibile più a lungo?



# Duello

Schiena contro schiena. Dieci passi, vi girate ed estraete la pistola. Siete abbastanza veloce? Riuscite a sparare per primo? Il computer stampa i numeri da 1 a 10 che rappresentano i 10 passi, si ferma poi stampa **ESTRAE LA PISTOLA ...** Dovete essere pronto a schiacciare un tasto (uno qualsiasi) appena queste parole appaiono sullo schermo. Se siete abbastanza veloci vincerete. Non premete tasti prima che appaia **ESTRAE LA PISTOLA** o perderete automaticamente.

## Come cambiare la velocità del gioco

Potete modificare il tempo a disposizione per reagire al messaggio cambiando l'ultimo numero nella linea 130; un numero minore vi concederà meno tempo (per il BBC cambiare il numero fra parentesi nella linea 140).

## Come rendere il gioco più difficile

Se cambiate il programma nel modo che segue, aggiungerete la possibilità di mancare il bersaglio:

- 1) cambiate 190 in 220 nella linea 140
- 2) Aggiungete queste linee



```
●▲■ s 10 PRINTCHR$(147)
      20 PRINT"IL DUELLO"
      30 PRINT"SIETE SCHIENA A SCHIENA"
      40 PRINT"FATE 10 PASSI..."
```

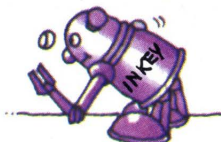
```
      50 FOR I=1TO10
      * 60 PRINT I;"..";
      ▲ s 65 FORJ=1TO300:NEXT J
      ▲ s ■ 70 NEXT I
```

```
      80 PRINT
      90 FOR I=1 TO RND(1)*1000
      100 NEXT I
```

```
●▲ s * 110 GET I$:IF I$<>""THEN GOTO 160
      120 PRINT"EGLI ESTRAE LA PISTOLA...";
```

```
*▲ s 130 FOR I=1TO50
●▲ s * 140 GET I$:IFI$<>""THENGOTO190
* 150 NEXT I
```

ALLA  
TASTIERA



```
160 PRINT"E SPARA!"
170 PRINT"SEI MORTO"
180 GOTO 210
190 PRINT"MA TU SPARI PER PRIMO."
200 PRINT"L'HAI UCCISO"
210 STOP
```

Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.

```
● 10 HOME
▲ * ■ s 10 CLS
      * 60 PRINT ;I;"..";
      ▲ s CANCELLA LA LINEA 65
      ▲ s 90 FOR I=1 TO RND*200
      ■ 90 FOR I=1 TO RND(0)*1000
      ● 110 IF PEEK(-16384)>127 THEN GOTO 160
      ▲ s 110 IF INKEY$<>"" THEN GOTO 160
      * 110 IF INKEY$(1)<>"" THEN GOTO 160
      ● 130 FOR I=1 TO 20
      ▲ s 130 FOR I=1 TO 5
      * CANCELLA LE LINEE 130, 150
      ● 140 IF PEEK(-16384)>127 THEN GOTO 190
      ■ ▲ s 140 IF INKEY$<>"" THEN GOTO 190
      * 140 IF INKEY$(40)<>"" THEN GOTO 190
```

```
SZ X 220 IF RND>.3 THEN GOTO 190
* ● 220 IF RND(1)>.3 THEN GOTO 190
■ 220 IF RND(0)>.3 THEN GOTO 190
230 PRINT "BUT YOU MISSED"
240 GOTO 90
```



## Come lavora il programma

Questo è un ciclo che fa stampare un numero e due puntini ad ogni iterazione.

Un altro ciclo, questa volta per far ritardare il computer. Il computer ripete il ciclo un numero di volte dipendente dal valore di RND senza fare nulla.

Questa riga controlla che non bariate schiacciando un tasto prima che appaia **ESTRAE LA PISTOLA**.

Stampa il segnale per premere un tasto.

Controlla la tastiera per vedere se avete premuto un tasto; se sí salta alla linea 190. (Notate che la linea 140 è all'interno di un ciclo **FOR ... NEXT**. Questo permette al computer di controllare la tastiera un certo numero di volte per darvi una ragionevole possibilità di premere un tasto).

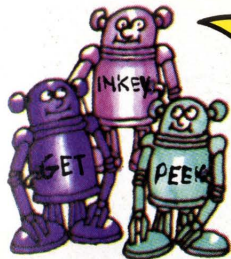
Stampa in caso di sconfitta (se avete sparato troppo tardi o stavate barando).

Stampa in caso di vittoria.

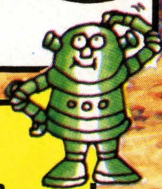
I cicli di ritardo possono essere scritti su una sola linea, come la linea 65, in tutti i computer escluso lo ZX81.



Un ciclo di ritardo extra per i computer piú veloci.



Notate il modo in cui i vari computer controllano la tastiera: il Vic usa **GET**, l'Apple deve usare il **PEEK**, gli altri **INKEY\$**.



### Rompicapo

Come si può fare in modo che anche il computer possa sbagliare?





# Battaglia nel deserto

L'ultima grande fortezza dell'esercito dei Robot al di fuori degli S.U.R.\* (\*Stati Uniti dei Robot) è nascosta all'interno delle rovine di un antico castello nel centro del deserto. Una flotta di aeronavi speciali è stata inviata a distruggerla e voi siete il comandante. La vostra aeronave controlla i cinque razzi rimasti.

Dovete calcolare attentamente la direzione e l'altezza del lancio prima di sparare. Il computer richiederà un angolo compreso fra  $-90^\circ$  (estrema sinistra) e  $+90^\circ$  (estrema destra) per la direzione ed un angolo compreso fra  $0^\circ$  (livello del terreno) e  $90^\circ$  (verticale perpendicolare al terreno) per l'alzo.

Quest'ultimo determinerà la distanza cui giungerà il razzo. Avete abbastanza mira da distruggere la fortezza dei robot?





## Come lavora il programma

```

10 PRINT "BATTAGLIA NEL DESERTO"
▲S ■ 20 LET T=INT(RND(1)*181)-90
▲S ■ 30 LET D=RND(1)
    
```



Sceglie un numero intero fra +90 e -90 per la direzione.

```

40 FORG=1T05
50 PRINT "DIREZIONE(-90 A 90) ?"
60 INPUT T1
70 PRINT "ALZO(0 A 90) ?"
80 INPUT B
    
```

Sceglie un numero fra 0 e 1 per la distanza del castello.

Legge i tentativi e li pone in A e B.

```

90 LET D1=SIN(2*(B/180*3.1416))
100 IF ABS(T-T1)<2 AND ABS(D-D1)<.05
    THEN GOTO 220
    
```

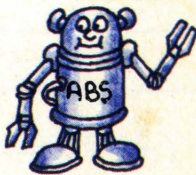
Usa l'alzo per calcolare la distanza cui il razzo finirà (il risultato sarà compreso fra 0 e 1).

Se la direzione che avete fornito era a meno di 2° da quella esatta e la distanza a meno di 0,05, allora avete colpito il bersaglio. Il programma salta alla linea 220 per annunciarlo.

```

110 PRINT "IL MISSILE E' ATTERRIATO ";
120 IF T1<T THEN PRINT "A SINISTRA ";
130 IF T1>T THEN PRINT "A DESTRA ";
135 PRINT "DELLA FORTEZZA"
    
```

Confronta l'angolo di direzione con quello scelto in linea 20 e stampa un messaggio conseguente.



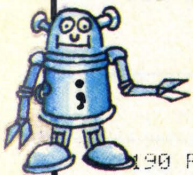
ABS prende il valore assoluto di un numero, cioè ne ignora il segno + o -.

Decide se stampare "e" confrontando la distanza cui è arrivato il razzo con il numero scelto in linea 30.

```

140 IF ABS(D-D1)>.05 AND T1<>T THEN PRINT "E ";
150 IF (D-D1)>.05 THEN PRINT "NON ABBASTANZA LONTANO";
160 IF (D-D1)<-.05 THEN PRINT "TROPPO LONTANO";
170 PRINT
180 NEXT G
    
```

Stampa un messaggio se il tiro è stato troppo lungo o troppo corto.



Un punto e virgola al termine di un'istruzione di stampa dice al computer di non passare ad una nuova riga nello stampare il prossimo elemento.

```

190 PRINT "DISASTRO-HAI FALLITO!"
200 PRINT "RITIRATA IGNOMINIOSA"
205 PRINT "VUOI SAPER LA POSIZIONE?"
206 INPUT R#:IF R#<0 THEN GOTO 210
207 PRINT "DISTANZA: "; PRINT D: PRINT "ALZO: ";
208 PRINT BB=90*3.1416*ATN(D/SQR(-D*D+1))
209 PRINT "DIREZIONE: "; PRINT T
210 END
220 PRINT "*800000000000M*"
230 PRINT "CE L'HAI FATTA!"
240 STOP
    
```

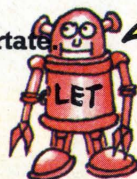
Stampa in caso di fallimento.

Stampa in caso di successo.

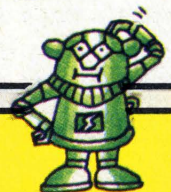
**Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.**

```

▲S 20 LET T=INT(RND*181)-90
■ 20 LET T=INT(RND(0)*181)-90
▲S 30 LET D=RND
■ 30 LET D=RND(0)
    
```



Solo per lo ZX81 è obbligatorio LET, si può tralasciare negli altri computer.

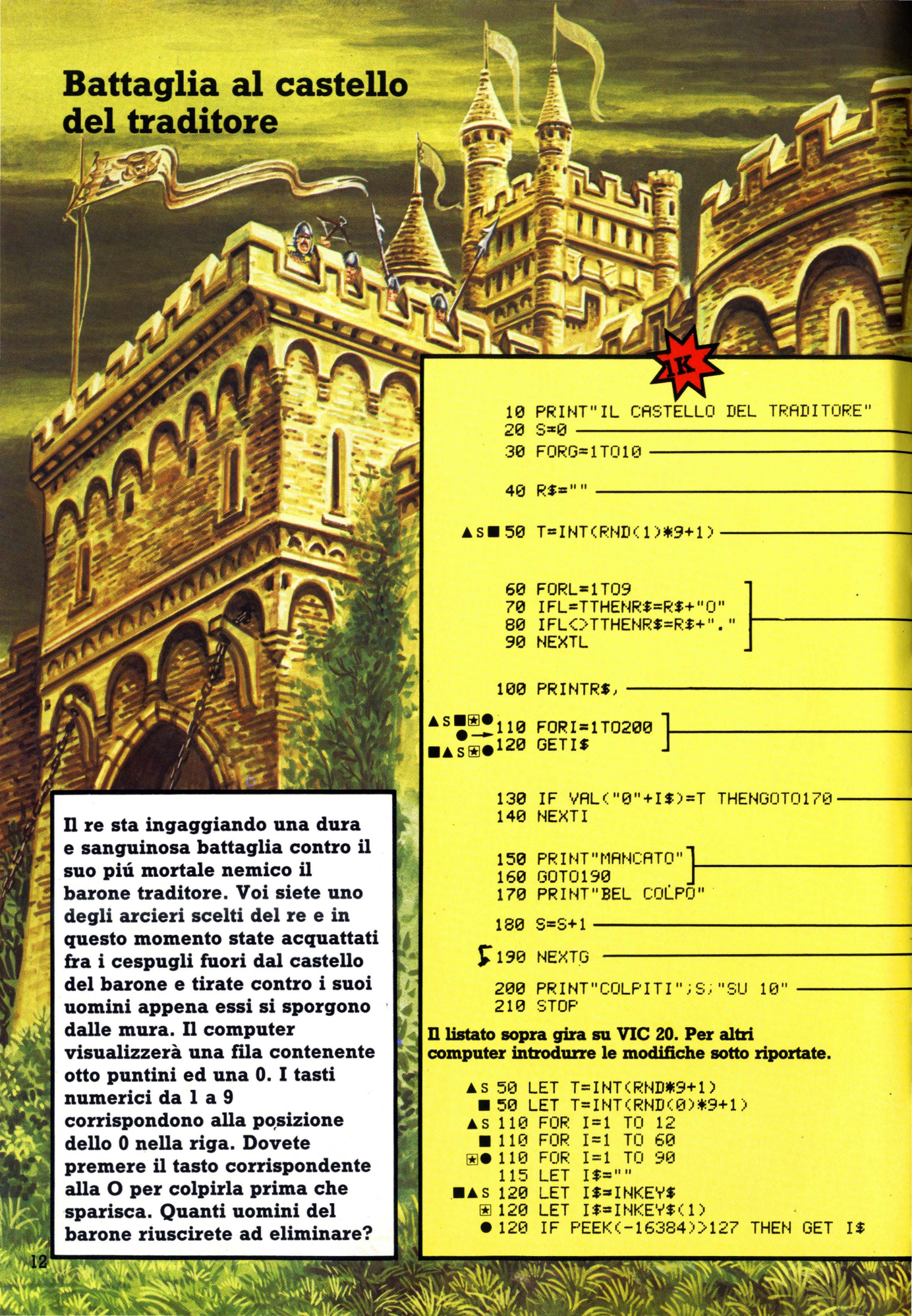


## Rompicapo

Come si può aggiungere la possibilità che i robot vi vedano e sparino prima che abbiate terminato i 5 tentativi?



# Battaglia al castello del traditore



Il re sta ingaggiando una dura e sanguinosa battaglia contro il suo piú mortale nemico il barone traditore. Voi siete uno degli arcieri scelti del re e in questo momento state acquattati fra i cespugli fuori dal castello del barone e tirate contro i suoi uomini appena essi si sporgono dalle mura. Il computer visualizzerà una fila contenente otto puntini ed una 0. I tasti numerici da 1 a 9 corrispondono alla posizione dello 0 nella riga. Dovete premere il tasto corrispondente alla 0 per colpirla prima che sparisca. Quanti uomini del barone riuscirete ad eliminare?



```
10 PRINT"IL CASTELLO DEL TRADITORE"
20 S=0
30 FORG=1TO10
40 R$=""
```

```
▲s■ 50 T=INT(RND(1)*9+1)
```

```
60 FORL=1TO9
70 IFL=TTHENR$=R$+"0"
80 IFL<>TTHENR$=R$+"."
90 NEXTL
```

```
100 PRINTR$,
```

```
▲s■ 110 FORI=1TO200
● 120 GETI$
```

```
130 IF VAL("0"+I$)=T THENGOTO170
140 NEXTI
```

```
150 PRINT"MANCATO"
160 GOTO190
170 PRINT"BEL COLPO"
```

```
180 S=S+1
```

```
190 NEXTG
```

```
200 PRINT"COLPITI";S;"SU 10"
210 STOP
```

Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.

```
▲s 50 LET T=INT(RND*9+1)
■ 50 LET T=INT(RND(0)*9+1)
▲s 110 FOR I=1 TO 12
■ 110 FOR I=1 TO 60
☒● 110 FOR I=1 TO 90
115 LET I$=""
■▲s 120 LET I$=INKEY$
☒ 120 LET I$=INKEY$(1)
● 120 IF PEEK(-16384)>127 THEN GET I$
```



## Come rendere il gioco piú veloce o piú lento

Se il computer lavora troppo velocemente o troppo lentamente in questo gioco; potete rimediarvi cambiando l'ultimo numero nella linea 110: un numero minore rende il gioco piú veloce.



## Rompicapo

Come si può cambiare il programma in modo da avere due tipi di bersaglio: una O (un soldato semplice del barone) che vale 1 punto o una S (uno dei soldati scelti) che vale 5 punti?

## Come lavora il programma

Azzerare il punteggio all'inizio del gioco.

Inizia un ciclo che permette di effettuare 10 tiri.

Inizializza una stringa vuota R\$.

Sceglie un numero fra 1 e 9 e lo pone in T.

Effettua nove volte un ciclo aggiungendo ogni volta un carattere a R\$. Il carattere è 0 per la posizione corrispondente al numero in T ed un puntino per le altre.

Stampa la stringa ottenuta.

Esegue un certo numero di volte un ciclo per controllare se state schiacciando un tasto (fine ciclo: linea 140).

Controlla se state premendo il tasto giusto, nel qual caso salta a 170.

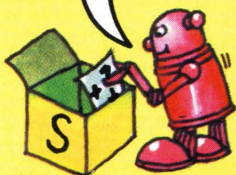
Stampa nel caso abbiate schiacciato il tasto sbagliato oppure fuori tempo massimo, poi salta a 190.

Aumenta di uno il punteggio.

Rimanda il computer alla linea 30 per un altro tiro.

Stampa il punteggio dopo 10 tentativi.

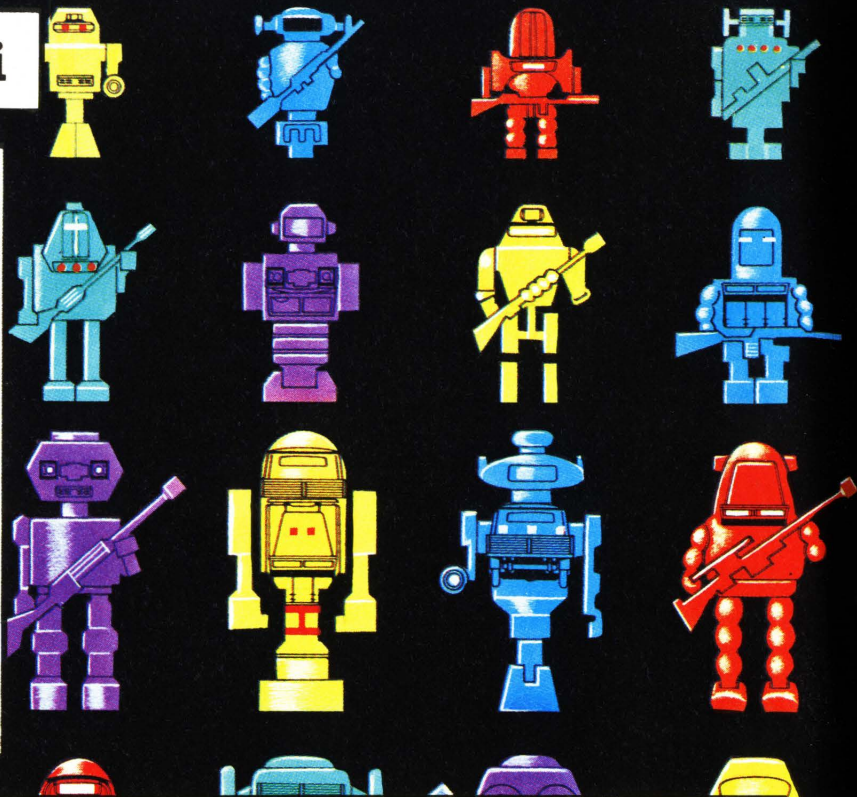
Notate come funziona il punteggio in questo gioco (linee 20 e 180). Potete cercare di adattarlo ad altri programmi.





# I Robot Invasori

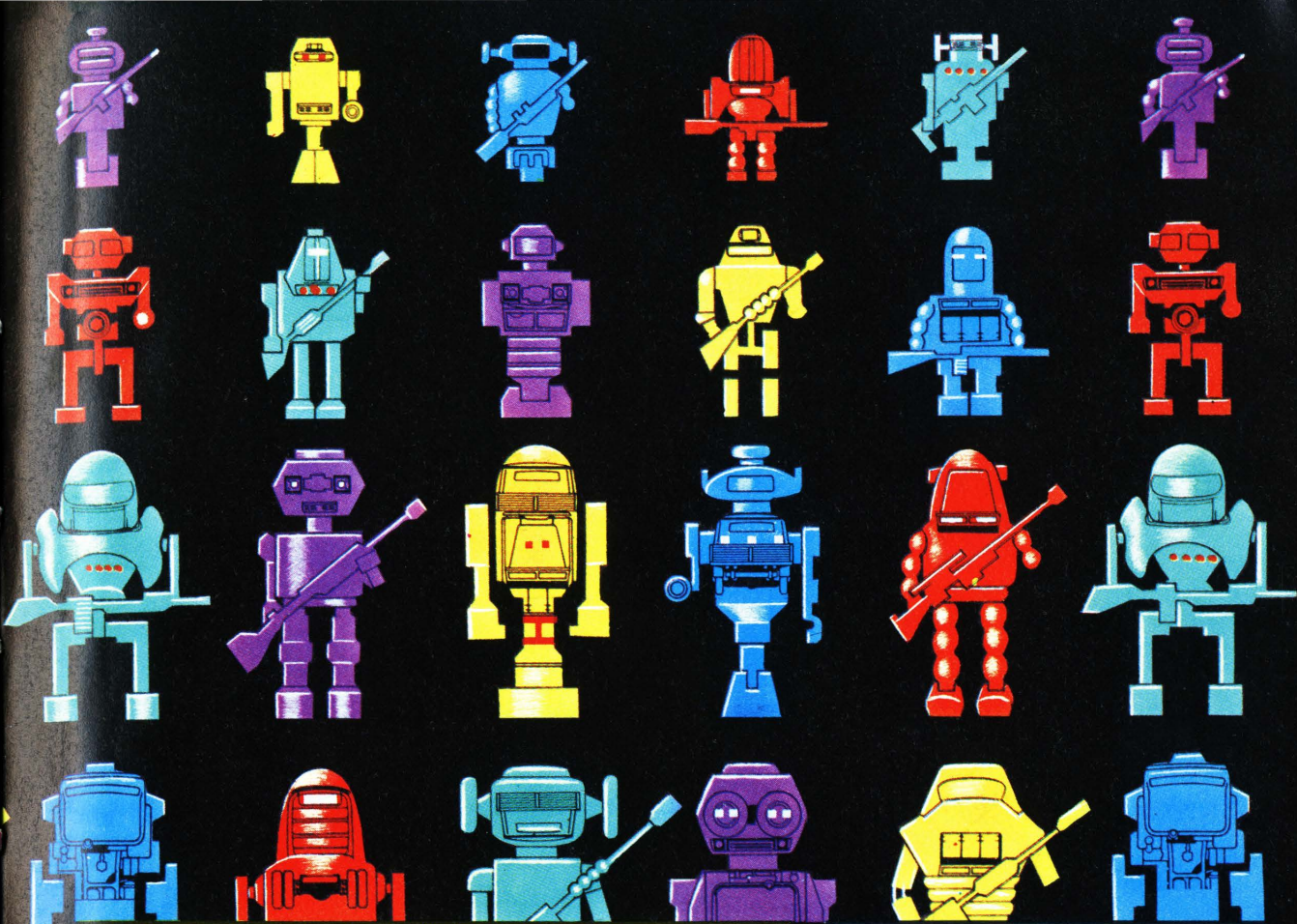
Dovete agire velocemente. Robot invasori di ogni tipo si stanno avvicinando. Avete a disposizione molte armi, ma per ogni tipo di robot si deve scegliere esattamente quella adatta per disattivarlo. I simboli in codice di ogni robot lampeggeranno sullo schermo. Premete velocemente il tasto corrispondente a quel simbolo — ma attenzione: per alcuni servirà anche il tasto SHIFT — e vedete quanti riuscirete a disattivarne.



## Come lavora il programma

10 PRINT "ROBOT INVADERS"	
20 H=0	Azzerare il punteggio all'inizio del gioco.
30 FORT=1 TO 25	Inizio del ciclo di 25 tentativi.
▲■ 40 FOR I=1 TO INT(RND(1)*300+200)	] Ritardo casuale.
50 NEXT I	
▲■ 60 A=INT(RND(1)*20)	] Sceglie i numeri che determinano la posizione sullo schermo.
▲■ 70 D=INT(RND(1)*15)	
▲■ 80 P#=CHR\$(INT(RND(1)*58+33))	Sceglie un carattere della tastiera.
▲◆■● 90 PRINT CHR\$(147)	Cancello lo schermo.
100 FOR J=0 TO D	] Muove il cursore verticalmente, una linea per volta, fino a raggiungere la linea D (scelta nella linea 70).
110 PRINT	
120 NEXT J	
130 PRINT TAB(A);P#	Muove il cursore orizzontalmente di A spazi e stampa il carattere scelto in linea 80.
● S ▲ 140 FOR I=1 TO 150	] Controlla la tastiera un certo numero di volte per vedere se state premendo un tasto, se è quello esatto e poi salta alla linea opportuna a seconda del risultato del controllo.
▲■ S ◆ 150 GET R#	
160 IFR#=P# THEN GOTO 210	
170 IFR#<>" " THEN GOTO 190	
180 NEXT I	
190 PRINT "MANCATO"	] Aumenta di uno il punteggio.
200 GOTO 230	
210 PRINT "COLPITO"	
220 H=H+1	
230 NEXT T	Procede ad un altro tentativo.
▲◆■● 240 PRINT CHR\$(147)	] Stampa il punteggio dopo 25 tentativi.
250 PRINT "PUNTEGGIO";H;"/25"	
260 STOP	





Il listato riportato gira su VIC 20. Per altri computer introdurre le modifiche qui sotto.

```

▲ s 40 FOR I=1 TO INT(RND*30+20)
  ■ 40 FOR I=1 TO INT(RND(0)*300+200)
▲ s 60, 70 CAMBIARE RND(1) CON RND
  ■ 60, 70 CAMBIARE RND(1) CON RND(0)
▲ 80 LET P#=CHR$(INT(RND*53+11))
  ■ 80 LET P#=CHR$(INT(RND(0)*58+33))
  s 80 LET P#=CHR$(INT(RND*58+33))
  ● 90, 240 HOME
▲*■ s 90, 240 CLS
  s ▲ 140 FOR I=1 TO 15
    ● 145 R#=""
  ■▲ s 150 LET R#=INKEY$
    * 150 R#=INKEY$(1)
    ● 150 IF PEEK(-16384)>127 THEN GET R#

```

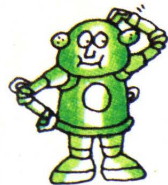
### Come rendere il gioco piú veloce

Man mano che diverrete esperti in questo gioco, probabilmente vorrete accelerarlo. Potete farlo cambiando l'ultimo numero della linea 410 in uno minore.

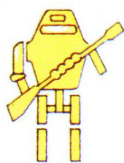
### Come muovere il cursore

Forse il vostro computer ha un modo piú semplice di muovere il cursore in un particolare punto dello schermo (vedi linee 110-130): controllate il manuale.

### Rompicapo



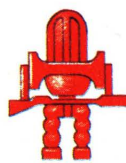
Questi sono i robot piú pericolosi.



**U**  
Mechans



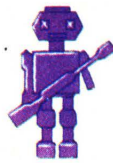
**V**  
Tintroids



**W**  
Scrapions



**X**  
Wireheads



**Y**  
Steeleyes

Come si può modificare il programma in modo da avere 100 punti per ognuno di questi e solo 10 per qualsiasi altro?



## Arma segreta

Se riuscite a distruggere il deposito principale delle parti di ricambio dei robot, che è situato sottoterra nei deserti orientali degli S.U.R., potreste danneggiare considerevolmente l'offensiva dei robot. Avete una nuova arma segreta, ancora ignota ai robot, che può penetrare attraverso le rocce vaporizzando qualsiasi cosa trovi sulla traiettoria; ma il deposito è ben nascosto. Tutto ciò che potete fare è sparare alla cieca e sperare di colpire vicino al deposito. Il computer vi chiederà un livello di difficoltà (il minimo è 4) quindi le coordinate X ed Y del tentativo (da immettere separatamente, premendo RETURN, NEWLINE o ENTER dopo ciascuna).

Leggete attentamente il programma se volete avere un indizio sui possibili valori di X ed Y.







### Come lavora il programma

```

■*▲S● 10 PRINTCHR$(147)
        20 PRINT"ARMA SEGRETA"
        30 PRINT"DIFFICOLTA'?"
        40 INPUT D
        50 IF D<4 THEN GOTO 30

▲S■ 60 X=INT(RND(1)*D+1)
▲S■ 70 Y=INT(RND(1)*D+1)
        80 FOR G=1 TO D+5

        90 PRINT" TENTATIVO PER X ED Y "
        100 INPUT X1
        110 INPUT Y1

        120 Z=SQR((X-X1)*(X-X1)+(Y-Y1)*(Y-Y1))

        130 IF Z=0 THEN GOTO 200
        140 IF Z<3 THEN PRINT"VICINO"
        150 IF Z>3 THEN PRINT"NEPPURE SFIORATO"

        160 NEXT G

        170 PRINT" I ROBOT TI HANNO VISTO "
        180 PRINT" AGGHHHHHHHH..... "

        190 STOP

        200 PRINT" L'HAI DISTRUTTO IN "
        210 PRINT G;" TENTATIVI "

        220 STOP

```

Riceve il livello di difficoltà, lo pone in-D e controlla che non sia inferiore a 4.

Sceglie i valori di X ed Y.

Inizio di un ciclo che offre un certo numero di tentativi a seconda della difficoltà richiesta.

Legge il tentativo e ne pone i valori in X1 e Y1.

Calcola la distanza fra il tiro ed il bersaglio e la mette in Z.



SQR calcola la radice quadrata.

Controlla Z per vedere a che distanza era e stampa un messaggio (se Z=0 avete vinto).

Fine del ciclo, inizia un nuovo tentativo.

Stampa nel caso terminino tutti i tentativi.

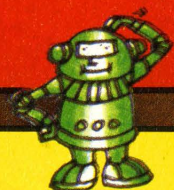
Stampa in caso di successo.

Il listato sopra riportato gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.

```

■*▲S 10 CLS
        ● 10 HOME
        ▲S 60 LET X=INT(RND*D+1)
        ■ 60 LET X=INT(RND(0)*D+1)
        ▲S 70 LET Y=INT(RND*D+1)
        ■ 70 LET Y=INT(RND(0)*D+1)

```



### Rompicapo.

Come si può aggiungere il seguente sistema di punteggio: 1 punto per ogni tiro vicino 10 punti per ogni centro?



# Fuga

I robot vi hanno catturato, hanno preso la vostra arma e vi hanno chiuso in cella. Improvvisamente ricordate di avere ancora l'orologio sonar, che può essere usato per produrre suoni di qualunque frequenza. Se vi riuscisse di trovare la frequenza di risonanza dei guardiani robot, essi dovrebbero vibrare fino a cadere a pezzi.

Però fate attenzione a non usare frequenze troppo basse, altrimenti l'edificio vibrerà e crollerà. Se invece saranno troppo elevate, vi verrà un tale mal di testa che dovrete smettere. Riuscirete a sfuggire agli orrori della prigione dei robot? (leggete attentamente il programma per avere un indizio sull'intervallo di frequenze da tentare).



```
■□▲● s 10 PRINT"J"  
20 PRINT"FUGA"  
▲ s ■ 30 F=INT(RND(1)*100+1)  
  
40 L=1 ]  
50 H=1 ]  
  
60 FORG=1T05  
  
70 PRINT"TENT.?" ]  
80 INPUT F1  
  
90 IFABS(F-F1)<5THENGOTO290  
  
100 IFF-F1>40THENGOTO170  
  
110 IFF1-F>40THENGOTO230  
120 PRINT"NESSUN EFFETTO"  
130 NEXTG  
140 PRINT"CI HAI MESSO TROPPO"  
150 PRINT"LA FREQUENZA ERA":F  
160 STOP  
  
170 IFL=2THENGOTO210  
180 PRINT"TROPPO BASSA...ATTENTO"  
190 L=2  
200 GOTO130  
210 PRINT"L'EDIFICIO E' CROLLATO"  
220 STOP  
  
230 IFH=2THENGOTO270  
240 PRINT"TROPPO ALTO...OUCH"  
250 H=2  
260 GOTO130  
270 PRINT"HAI MAL DI TESTA LASCIA STARE"  
280 STOP  
290 PRINT"CE L'HAI FATTA"  
300 STOP
```

## Come lavora il programma

Sceglie un numero fra 1 e 100 per la frequenza dei robot e lo pone in F.

Pone l in L e in H; queste variabili sono usate se andate troppo in alto o in basso con le frequenze — vedi linee 170-190 e 230-250.

Inizia un ciclo che permette di avere 5 tentativi.

Riceve un tentativo e lo pone in F1.

Controlla se il tentativo differisce da F per meno di 5; in questo caso salta a 290 e stampa CE L'HAI FATTA.

Salta a 170 se la frequenza che avete tentato è minore di F di almeno 40.

Salta a 230 se la frequenza che avete tentato è maggiore di F di almeno 40.

Stampa nel caso il tentativo sia diverso da F per meno di 40 e passa al tentativo seguente. Dopo l'ultimo stampa la risposta.

Controlla il valore di L. Quando questa parte del programma viene raggiunta la prima volta il valore di L è 1: il computer procede a stampare un avviso, cambia il valore di L in 2 e passa al tentativo seguente. La prossima volta che il programma arriverà alla linea 170 salterà direttamente alla linea 210 per annunciare che avete perso.

Queste linee controllano H allo stesso modo: la prima volta per avvisare che usate frequenze troppo alte e la seconda volta che avete perso.

Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.

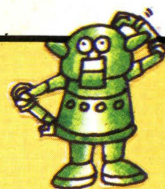
```
▲□■ s 10 CLS  
● 10 HOME  
▲ s 30 LET F=INT(RND*100+1)  
■ 30 LET F=INT(RND(0)*100+1)
```





## Rompicapo

Ognuna delle tre guardie robot ha una diversa frequenza di risonanza; non potete scappare finché non le avete trovate tutte e tre. Come si può cambiare il programma per far ciò?



## Come rendere il gioco piú difficile

Cambiate il 5 nella linea 90 con un numero minore; ciò significa che dovete avvicinarvi di piú a F per vincere. Potete anche ampliare il possibile intervallo di F cambiando il 100 nella linea 30 in un numero maggiore.





# Caccia al pirata

Siete alla caccia del pirata dell'aria. Quando egli vi precede accelerate, quando si lascia superare rallentate; dovete cercare di accostarvi a lui e poi sparare, sperando che non riesca lui per primo a colpirvi.

Usate la lettera A per accelerare, D per decelerare ed F per aprire, il fuoco. Il computer vi dirà la vostra velocità e la vostra posizione relativa rispetto al pirata. Dovrete essere veloce nel premere i tasti appena battuto RUN. Continuate a premere A e D finché non sarete al suo livello e poi sparate.



```

■□▲S● 10 PRINT"□"
        20 PRINT"CACCIA AL PIRATA"
▲S■ 30 V=INT(RND(1)*11-5)
    
```

## Come lavora il programma

Sceglie un numero fra -5 e +5 per la vostra velocità rispetto al pirata e lo pone in V.

```

▲S■ 40 S=INT(RND(1)*3+1)
    
```

Sceglie un numero per la distanza dal pirata e lo pone in S. All'inizio questo numero è negativo, ciò significa che siete alle sue spalle.

```

        50 IFABS(S)>20 THENGOTO230
■□▲S● 60 PRINT"□"
    
```

Controlla se la distanza da lui è maggiore di 20; in questo caso il computer salta a 230 per dirvi che l'avete perso di vista.

```

70 PRINT"SEI";
80 IFSC<0THENPRINT" DIETRO"
90 IFS>0THENPRINT" DAVANTI"
100 IFS=0THEN PRINT" DI FIANCO"
110 PRINT"STAI ANDANDO";
120 IFV>0THENPRINT"PIU' VELOCE"
130 IFV<0THENPRINT"PIU' LENTAMENTE"
140 IFV=0THENPRINT"ALLA STESSA VELOCITA'"
    
```

Controlla i valori di S e V stampa posizione e velocità rispetto a quelle del pirata.

```

●→
■▲S■□ 150 GETI$
        160 IFI$="A"THENV=V+1
        170 IFI$="D"THENV=V-1
        180 IFI$="F"AND S=0THENGOTO250
    
```

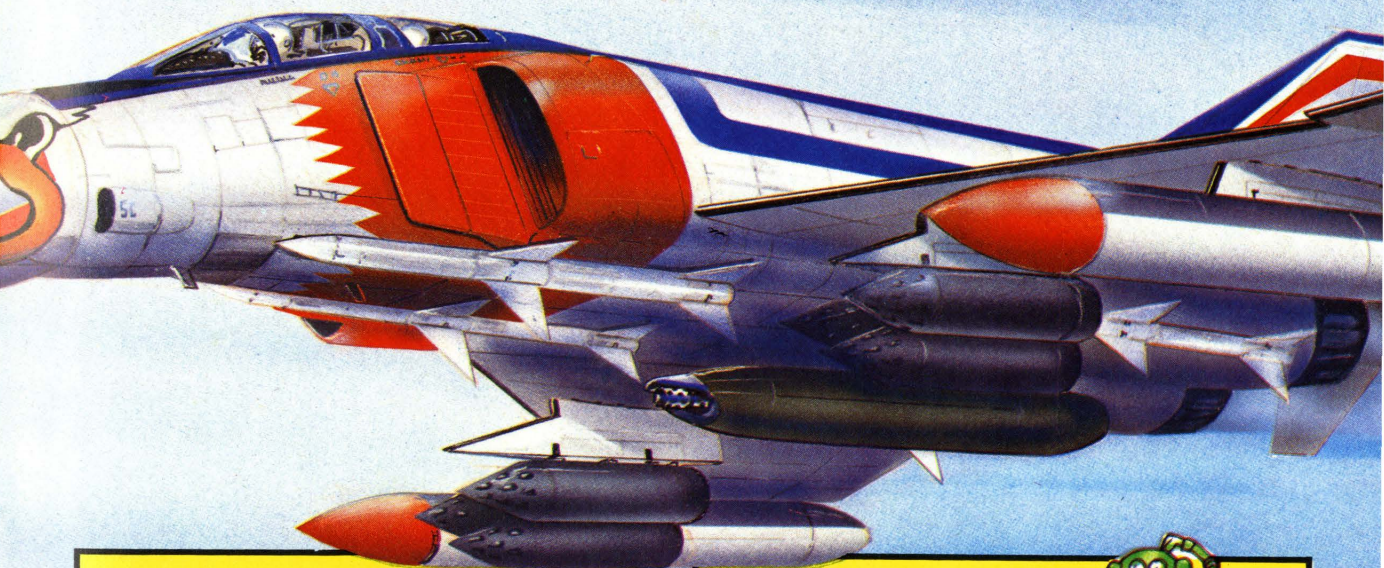
Controlla se state premendo un tasto e quale. Se è A la vostra velocità aumenta di uno; se è D diminuisce di 1. Se state premendo F ed S è uguale a zero allora salta a 250 (se S è diversa da 0 quando premete F non accade nulla).



190 S=S+V	_____	Calcola la nuova distanza relativa
▲s 200 FOR I=1 TO 200	] _____	Ciclo di ritardo.
210 NEXT I		
220 GOTO 50		
230 PRINT "PERDUTO DI VISTA"		
240 GOTO 330		Torna a 50 per ripetere.
		Viene eseguita se premete F quando siete al livello giusto; controlla se la velocità è minore di 2 (indipendentemente dal segno), nel qual caso salta a 290.
250 IF ABS(V)<2 THEN GOTO 290	_____	
		Stampa un messaggio se la velocità era maggiore di 2 quando avete sparato; torna a 40 per iniziare con un nuovo valore di S.
260 PRINT "SEI RIUSCITO SOLO A"	] _____	
270 PRINT "SPAVENTARLO"		
280 GOTO 40		
▲s ■ 290 IFRND(1)>.7 THEN GOTO 320	_____	Applica un test casuale per vedere se avete sparato prima del pirata.
		Stampa un messaggio a seconda del risultato del test della linea 290.
300 PRINT "L'HA I ABBATTUTO"	] _____	
310 GOTO 330		
320 PRINT "HA SPARATO PER PRIMO"		
330 STOP		

Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.

- |                             |  |
|-----------------------------|--|
| ■▲s 10, 60 CLS              | ● 145 I\$=""                           |
| ● 10, 60 HOME               | ■▲s 150 LET I\$=INKEY\$                |
| ▲s 30 LET V=INT(RND*11-5)   | ☒ 150 I\$=INKEY\$(1)                   |
| ■ 30 LET V=INT(RND(0)*11-5) | ● 150 IF PEEK(-16384)>127 THEN GET I\$ |
| ▲s 40 LET S=-INT(RND*3+1)   | ▲s 200 FOR I=1 TO 20                   |
| ■ 40 LET S=-INT(RND(0)*3+1) | ▲s 290 IF RND>.7 THEN GOTO 320         |
|                             | ■ 290 IF RND(0)>.7 THEN GOTO 320       |



### Come rendere il gioco piú facile

Potreste trovare questo gioco piuttosto difficile. Per facilitarlo un po', aggiungete queste due linee: potrete vedere sullo schermo le posizioni relative dei due aerei.

```
195 IF ABS(S)>10 THEN GOTO 200
196 PRINT TAB(W/2);"LUI"
197 PRINT TAB(S+W/2);"TU"
```

Sostituite w con la larghezza del vostro schermo.

### Rompicapo



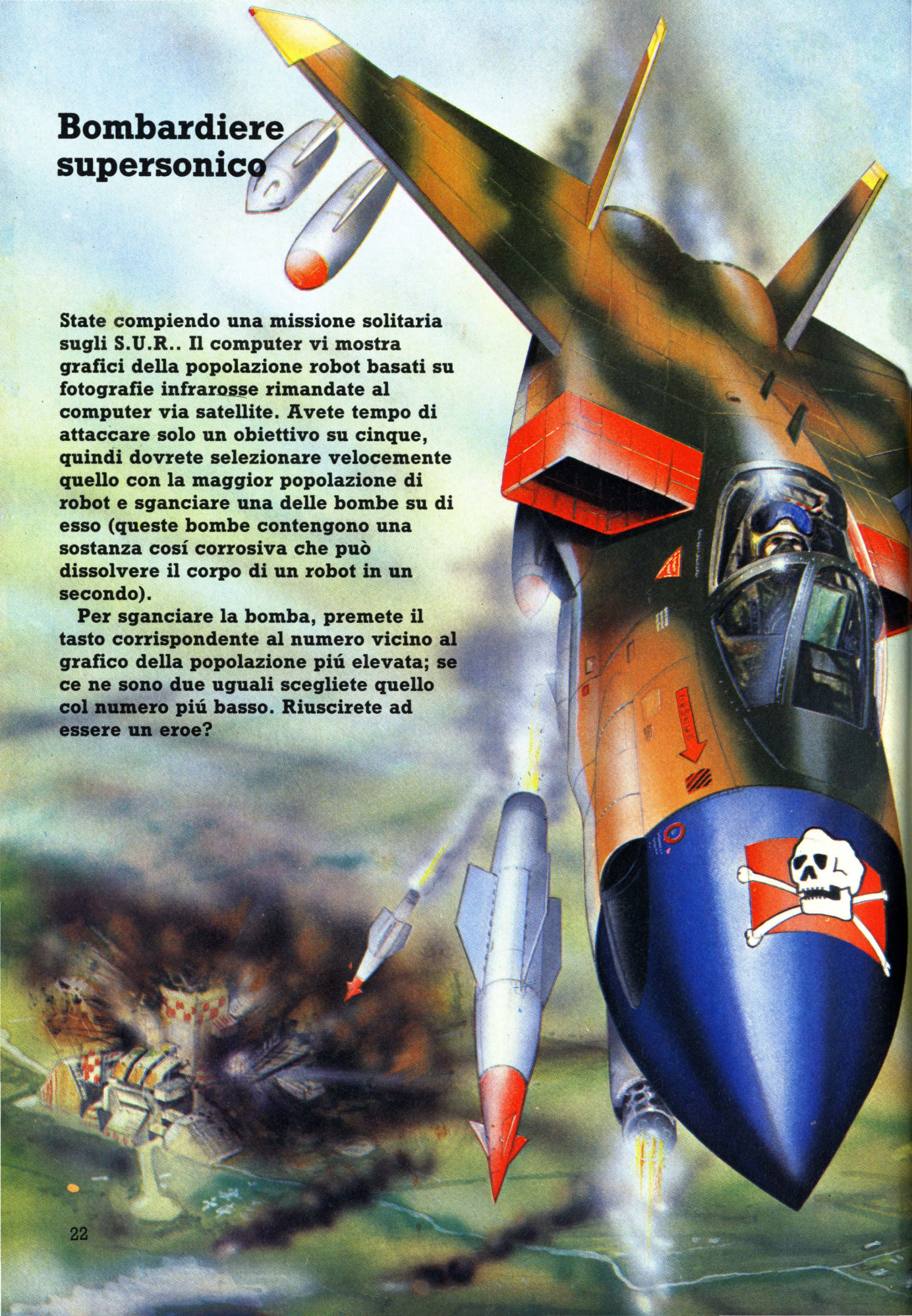
Il test casuale nella linea 290 è in vostro favore: come si può cambiare in modo che le probabilità di vittoria siano pari?



# Bombardiere supersonico

State compiendo una missione solitaria sugli S.U.R.. Il computer vi mostra grafici della popolazione robot basati su fotografie infrarosse rimandate al computer via satellite. Avete tempo di attaccare solo un obiettivo su cinque, quindi dovrete selezionare velocemente quello con la maggior popolazione di robot e sganciare una delle bombe su di esso (queste bombe contengono una sostanza così corrosiva che può dissolvere il corpo di un robot in un secondo).

Per sganciare la bomba, premete il tasto corrispondente al numero vicino al grafico della popolazione più elevata; se ce ne sono due uguali scegliete quello col numero più basso. Riuscirete ad essere un eroe?





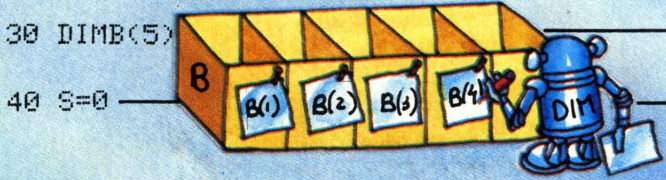


## Come lavora il programma

```

10 PRINTCHR$(147)
20 PRINT"BOMBARDIERE SUPERSONICO"

```



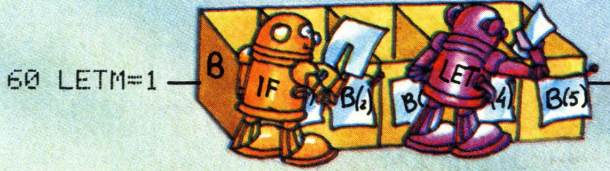
Predisporre B come variabile matrice; può contenere cinque valori differenti per volta, identificandoli con B(1), B(2), B(3), B(4), B(5).

```
40 S=0
```

Inizializza il punteggio a zero.

```
50 FORG=10TO1STEP-1
```

Inizia un ciclo di 10 tentativi, che in questo gioco viene percorso a rovescio (cioè: G=10 al primo tentativo, 9 al secondo e così via). Questo fa in modo che il ciclo di ritardo della linea 190 dia più tempo per i primi tentativi.



Inizializza a 1 il numero del grafico con popolazione maggiore.

```

70 FORI=1TO5
80 LETB(I)=INT(RND(1)*10+1)
90 IFB(I)>B(M)THENLETM=I
100 NEXTI

```

Sceglie 5 numeri e li pone nella matrice da B(1) a B(5). Sceglie il maggiore e cambia M con questo.

```

110 PRINTCHR$(147)
120 FORI=1TO5
130 PRINTI
140 FORJ=1TOB(I)
150 PRINT"***";
160 NEXTJ
170 PRINT
180 NEXTI

```

Stampa i 5 numero sullo schermo sotto forma di 5 righe di asterischi.

```

190 FORI=1TOG*30
200 GETI$
210 IFI$<>" "THENGOTO270
220 NEXTI
230 PRINT"TROPPO TARDI"
240 FORJ=1TO400
250 NEXTJ
260 GOTO280
270 IFVAL(I$)=MTHENS=S+1
280 NEXTG

```

Controlla se state premendo un tasto e in questo caso salta a 270.

Se non avete schiacciato in tempo stampa TROPPO TARDI e passa al tentativo successivo.

Controlla se stavate premendo il tasto giusto, nel qual caso aumenta di 1 il punteggio.

Passa al tentativo seguente.

```

290 PRINT"COLPITI";S;"SU 10"
300 PRINT"BERSAGLI AD ALTA DENSITA'"
310 IFS=10THENPRINT"SEI UN EROE"
320 IFS<10THENPRINT"HAI FALLITO"
330 STOP

```

Stampa il punteggio dopo 10 tentativi.

**Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.**

```

10, 110 CLS
10, 110 HOME
80 LET B(I)=INT(RND*10+1)
80 LET B(I)=INT(RND(0)*10+1)
190 FOR I=1 TO G*3

```

```

195 I$=""
200 LET I$=INKEY$
200 I$=INKEY$(1)
200 IF PEEK(-16384)>127 THEN GET I$
240 FOR J=1 TO 10

```

### Come cambiare la velocità del gioco

Per avere più possibilità di premere un tasto ogni volta, cambiate l'ultimo numero della linea 190 in uno maggiore. Man mano che diventerete più abile, diminuite il numero nella linea 190: fino a che numero riuscite ad arrivare vincendo?

### Rompicapo



**Come si può fare in modo che il computer offra più di 5 bersagli possibili alla volta?**



# Iceberg

Il vostro scafo è gravemente danneggiato e non avete armi. Mentre avanzate lentamente sulle acque pericolosamente cosparse di Iceberg, vi accorgete che una nave nemica vi sta seguendo. Stranamente essa rileva la vostra posizione, ma non quella degli iceberg quindi avete l'opportunità di indurla a scontrarsi con uno di essi.

Il computer stamperà una griglia indicante la posizione della vostra nave (Y), di quella del nemico (Z), e degli iceberg. Potete muovervi di uno spazio verso Nord, Sud, Est o Ovest ad ogni turno; il nemico avanza verso di voi lungo la rotta piú diretta (si può muovere anche in diagonale). Se vi muovete in una delle otto posizioni che circondano il nemico, sarete catturato e se colpirete un iceberg affonderete. Riuscirete a sfuggire?



## Come lavora il programma

Prepara la griglia; B è una matrice di DIMensioni 8 per 8.

N è il numero di iceberg; varia fra 4 ed 11 secondo il valore di RND.

23 sullo ZX e 42 sugli altri computer è il codice di "\*". Il computer esegue un ciclo N volte mettendo "\*" in posizioni casuali nella griglia.

Mette la nave nemica sulla griglia all'inizio del gioco. 63 (ZX81) e 90 (altri) sono i codici di Z. Controlla che la posizione non sia già stata assegnata ad un iceberg, altrimenti ne trova un'altra.

Queste linee fanno lo stesso delle precedenti per mettere la vostra nave sulla griglia (62 e 89 sono i codici di Y).

Stampa la griglia con le posizioni correnti delle navi e degli iceberg.

```
10 PRINT"ICEBERG"
20 DIMB(8,8)
```

```
▲S■ 30 N=INT(RND(1)*8+4)
```

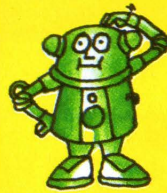
```
40 FORI=1TON
▲S■ 50 B(INT(RND(1)*8+1),INT(RND(1)*8+1))=42
60 NEXTI
```

```
▲S■ 70 SX=INT(RND(1)*8+1)
▲S■ 80 SY=INT(RND(1)*8+1)
90 IFB(SX,SY)≠0THENGOTO70
▲100 B(SX,SY)=90
```

```
▲S■ 110 YX=INT(RND(1)*8+1)
▲S■ 120 YY=INT(RND(1)*8+1)
130 IFB(YX,YY)≠0THENGOTO110
▲140 B(YX,YY)=89
```

```
■*▲S● 150 PRINTCHR$(147)
160 FORY=1TO8
170 FORX=1TO8
180 IFB(X,Y)≠0THENGOTO210
190 PRINTCHR$(B(X,Y));
200 GOTO220
210 PRINT".";
220 PRINT" ";
230 NEXTX
240 PRINT
250 NEXTY
```





## Rompicapo

Come si può ingrandire la griglia ed aggiungere gli iceberg?



260 B(YX,YY)=0

Cancella dalla griglia la vostra posizione in modo che se ne possa specificare una nuova.

270 PRINT"DIREZIONE(N,S,E,W) " ]

280 INPUTD\$

Mette in D\$ la vostra direzione.

290 YY=YY-(D\$="S"ANDYY<8)

300 YY=YY+(D\$="N"ANDYY<1)

310 YX=YX-(D\$="E"ANDYX<8)

320 YX=YX+(D\$="W"ANDYX<1)

Calcola la vostra nuova posizione, controllando che non superi il bordo della griglia.

▲ 330 IFB(YX,YY)=90THENGOTO500

▲ 340 IFB(YX,YY)=42THENGOTO600

Controlla la nuova posizione per vedere se vi siete scontrato con la nave nemica o con un iceberg, in questo caso salta più avanti nel programma per avvisarvi.

▲ 350 B(YX,YY)=89 ]

360 B(SX,SY)=0

Siete passato ad una posizione sicura dove viene posto il codice della nave.

370 SX=SX+SGN(YX-SX)

380 SY=SY+SGN(YY-SY)

Calcola la nuova posizione del nemico.

▲ 390 IFB(SX,SY)=89THENGOTO500

▲ 400 IFB(SX,SY)=42THENGOTO700

Controlla la nuova posizione del nemico per vedere se vi ha catturato o se ha urtato un iceberg, in questo caso salta più avanti nel programma per avvisarvi.

▲ 410 B(SX,SY)=90

Il codice del nemico è posto nella nuova posizione.

420 GOTO150

500 PRINT"SEI STATO PRESO"

510 GOTO800

600 PRINT"HAI COLPITO UN ICEBERG"

610 GOTO800

700 PRINT"SEI SALVO-LUI NE HA COLPITO UNO"

800 STOP

Fine di un turno, torna a 150 per iniziare uno nuovo.

**Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.**

- ▲ s 30,50,70,80,110,120 CAMBIARE RND(1) CON RND
- 30,50,70,80,110,120 CAMBIARE RND(1) CON RND(0)
- ▲ 50,340,400 CAMBIARE 42 IN 23
- ▲ 100,300,410 CAMBIARE 90 IN 63
- ▲ 140,350,390 CAMBIARE 89 IN 62

■ s 150 CLS

● 150 HOME

290 LET YY=YY+(D\$="S" AND YY<8)

300 LET YY=YY-(D\$="N" AND YY<1)

▲ s ● 310 LET YX=YX-(D\$="E" AND YX<8)

320 LET YX=YX+(D\$="W" AND YX<1)

I computers possono stabilire se una risposta è vera o falsa. Se è falsa le assegnano valore 0; se è vera alcuni computers assegnano valore +1 altri -1. Per questo le linee 290-320 vanno cambiate per alcuni computer.





## Il muro del mago

Il muro costruito dal mago per circondare la sua fortezza segreta non è un muro comune. Le sue pietre sono persone rese tali dallo sguardo malvagio del mago ed inoltre esso si può muovere. Se riuscite ad aprire una breccia in esso, grazie a catapulte che scagliano massi giganteschi, distruggerete i poteri magici del mago trasformando di nuovo le pietre in persone. Ma ci sono dei problemi: il mago ha preparato un incantesimo che ferma i missili in volo e li rimanda da dove sono partiti, a volte egli lo usa ... Se non avrete successo diverrete un'altra pietra nel muro del mago!!!!

```
★■●▲S 10 PRINTCHR$(147)
      20 PRINT"IL MURO DEL MAGO"
      30 PRINT
      40 PRINT"VUOI AIUTO?"
      50 INPUT I$
      ▲S 60 IFLEFT$(I$,1)="S"THENGOSUB740
      70 PRINT"DIFFICOLTA'?"
      80 PRINT"(5=FACILE,1=DIFFICILE)"
      90 INPUTQ
      100 DIMW(8,8)
      110 DIME(8)
      120 FORY=1TO8
      130 FORX=1TO4
      ▲S 140 LETW(X,Y)=29
      150 NEXTX
      160 NEXTY
      170 Z=0
      180 FORC=1TO3
      190 LETD=INT(RND(1)*80+21)
      200 GOSUB430
      210 IFZ=1THENGOTO1150
      220 PRINT"SEI A ";D;" LEGHE DI DISTANZA"
      230 W=INT(RND(1)*41)-20
      240 IFW=0THENPRINT"ASSENZA DI VENTO"
      250 IFW<0THENPRINT"VENTO DA DESTRA"
      260 IFW>0THENPRINT"VENTO DA SINISTRA":
      270 PRINT"ALZO (1-90)"
      280 INPUT A
      290 PRINT"VELOCITA'"
```



```

300 INPUT V
310 A=A/180*3.1416
320 H=TAN(A)*(D-W)-(5*(D-W)^2)/(V*COS(A))^2
330 H=INT(H/Q)
340 IFH>0ANDH<9THENGOSUB540
350 IFH<1THENPRINT"TIRO TROPPO VICINO"
360 IFH>8THENPRINT"TIRO TROPPO LONTANO"
370 IFRND(1)>.2THENGOTO200
380 PRINT"IL MURO SI E' MOSSO..."
390 NEXT C
400 PRINT"SEI STATO TRASFORMATO"
410 PRINT"IN PIETRA"
420 STOP
*▲S 430 FORI=1TO200
440 NEXTI
*■●▲S 450 PRINTCHR$(147)
460 PRINT
470 FORY=8TO1STEP-1
480 FORX=1TO8
▲S 490 PRINTCHR$(W(X,Y)+32);
500 NEXTX
510 PRINT
520 NEXTY
530 RETURN
540 GOSUB650
550 W(E(H)-1,H)=0
560 IFV*COS(A)>50 THENW(E(H)-2,H)=0
570 IFRND(1)>.5ANDH<1THENW(E(1),1)=29
580 IFRND(1)>.5ANDH>5ANDH<8THENW(E(H+1)-1,H+1)=0
590 IFH>1THENGOTO630
600 FORY=2TO8
610 IFRND(1)<.5THENW(E(Y)-1,Y)=0
620 NEXTY
630 GOSUB650
640 RETURN
650 FORY=1TO8
660 X=1
670 IFW(X,Y)=0ORX=8THENGOTO700
680 X=X+1
690 GOTO670
700 E(Y)=X
710 IFX=1THENZ=1
720 NEXTY
730 RETURN

```

Sostituire \*\* nella linea 320 con ^ per BBC ed Apple o † per il VIC, TRS 80 e Spectrum.



## Il muro del mago (seguito)

```
731 REM*****
732 REM** IL TESTO PUO' ESSERE LEG***
733 REM**GERMENTE DIVERSO DA QUELLO***
734 REM**DELLA TRADUZIONE SCRITTA. ***
735 REM*****
740 PRINT"STAI ATTACCANDO L'ULTIMA"
750 PRINT"FORTEZZA DEL FAMIGERATO "
760 PRINT"MAGO CHE SI NASCONDE    "
770 PRINT"DIETRO UN MURO DI PIETRA"
780 PRINT"SENZA FINE,ED OGNI PIETRA"
790 PRINT"E' UNA DELLE SUE VITTIME"
800 PRINT"SOLO TU PUOI ATTACCARLO"
810 PRINT"E LIBERARLI DAL MALVAGIO"
820 PRINT"INCANTESIMO.DEVI ABBATTERE"
830 PRINT"IL MURO USANDO CATAPULTE"
840 PRINT"MA ATTENZIONE:IL MAGO HA"
850 PRINT"IL POTERE DI MUOVERE IL"
860 PRINT"MURO AVANTI E INDIETRO E"
870 PRINT"OCCASIONALMENTE ANCHE DI "
880 PRINT"RESPINGERE I COLPI VERSO"
890 PRINT"DI TE."
920 GOSUB1110
930 PRINT"DOPO OGNI TIRO,TI VIENE"
940 PRINT"MOSTRATA LA SEZIONE DEL"
950 PRINT"MURO,EVIDENZIANDO I DANNI"
960 PRINT"SUBITI DA ESSO"
970 PRINT"NOTA CHE CI SONO ALCUNE"
980 PRINT"PIETRE CHIAVE CHE PRODUCONO"
990 PRINT"DANNI ENORMI;INOLTRE PIU'"
1000 PRINT"VELOCE E' IL MASSO IN "
1010 PRINT"DIREZIONE ORIZZONTALE, "
1020 PRINT"MAGGIORE E' IL DANNO"
1030 PRINT"CAUSATO,RIUSCIRAI A"
1040 PRINT"SCONFIGGERE IL MAGO IN"
1050 PRINT"TEMPO PER SALVARE LE"
1060 PRINT"MIGLIAIA DI ANIME"
```



```

1070 PRINT"PRIGIONIERE???????"
1090 GOSUB1110
1100 RETURN
1110 PRINT"PREMI UN TASTO"
★▲S●1120 GETI$:IFI$=""THENGOTO1120
★■●▲S1130 PRINTCHR$(147)
1140 RETURN
1150 PRINT"SEI RIUSCITO AD APRIRE"
1160 PRINT"UNA BRECCIA NEL MURO DEL"
1170 PRINT"MAGO-HAI BATTUTO I SUOI "
1180 PRINT"POTERI MAGICI E LIBERATO"
1190 PRINT"LE SUE VITTIME."
1200 STOP

```

**Il listato sopra gira su VIC 20. Per altri computer introdurre le modifiche sotto riportate.**

```

CAMBIARE RND(1) CON RND
CAMBIARE RND(1) CON RND(0)
★■▲S 10,450,1130 CLS
● 10,450,1130 HOME
▲S 60 IF I$(1)="S" THEN GOSUB 740
▲S 140 LET W(X,Y)=20
▲S 430 FOR I=1 TO 30
★ 430 FOR I=1 TO 1000
▲S 490 PRINT CHR$(W(X,Y));
▲S 1120 IF INKEY$="" THEN GOTO 1120
★ 1120 I$=GET I$
● 1120 GET I$

```

### **Cambiamenti che si possono apportare al gioco**

Potete rendere il muro piú spesso cambiando le linee seguenti:

Linea 100: cambiate il primo numero in uno maggiore, ad es. 10.

Linea 130: sottraete 4 dal numero usato nella linea 100 e mettete il risultato alla fine della linea 120 (ad es. in questo caso usa 6)

Linea 480: cambiate l'ultimo numero con il numero usato nella linea 100 (ancora 10 in questo caso)

Se rendete il muro piú robusto probabilmente avrete bisogno di piú tempo: per ottenere ciò cambiate l'ultimo numero nella linea 180 in uno maggiore.



# Missile

Questo gioco è diverso dagli altri in questo libro perché usa la grafica. Dal momento che i computers variano molto nel loro funzionamento grafico, c'è un programma separato per ciascuno di essi. Leggete le istruzioni per capire come si svolge il gioco, poi cercate nelle pagine seguenti la versione per il vostro computer.



## Come giocare a Missile!

Avete tre basi missilistiche, ognuna in grado di lanciare un missile. Quando vedete avvicinarsi un aereo dovete giudicare la sua altezza e velocità e sparare i missili uno alla volta contro di esso.

I missili vengono lanciati premendo un tasto qualsiasi: la prima volta che si preme un tasto partirà il missile di sinistra, la seconda quello di centro e la terza quello di destra.

Contate quanti aerei nemici riuscite ad abbattere.

### Missile!: versione TRS-80

```
10 CLS
20 DIM Y(3),F(3)
30 N=1
40 PS=INT(RND(0)*3+1)
50 P=INT(RND(0)*36+5)
60 GOSUB 400
70 FOR I=PS TO 100 STEP PS
80 GOSUB 300
90 F$=INKEY$
100 IF F$="" OR N>3 THEN 130
110 F(N)=1
120 N=N+1
130 FOR J=1 TO 3
140 RESET (32*J,46-Y(J))
150 IF F(J)=0 OR Y(J)>45 THEN 190
160 Y(J)=Y(J)+1
170 IF POINT(32*J,46-Y(J)) THEN 230
180 SET (32*J,46-Y(J))
190 NEXT
200 NEXT
210 PRINT@0,"MANCATO"
220 END
230 PRINT@0,"COLPITO"
240 END
300 RESET(I-PS,P) : RESET (I-PS+1,P)
310 SET (I,P) : SET (I+1,P)
320 RETURN
400 FOR J=1 TO 3
410 SET (J*32,47)
420 SET (J*32+1,47)
430 NEXT
440 RETURN
```



## Missile!: versione BBC

```
10 MODE 5
20 VDU 23,224,224,160,144,144,143,128,128,255
30 VDU 23,225,0,0,0,0,240,12,2,255
40 VDU 23,226,16,56,84,16,16,16,0,0
50 VDU 23,227,0,0,0,8,8,8,8,60
60 DIM Y(3),F(3)
70 N=1 : MS=16
80 PS=RND(20)+10
90 P=RND(500)+400
100 PROCDISPLAY
110 FOR I=PS TO 1100 STEP PS
120 PROCPLANE(I-P,P,0) : PROCPLANE (I,P,3)
130 F#=INKEY$(0)
140 IF F#="" OR N>3 THEN 170
150 F(N)=TRUE
160 N=N+1
170 FOR J=1 TO 3
180 IF NOT F(J) THEN 240
190 PROCMISSILE(J,0)
200 Y(J)=Y(J)+MS
210 IF Y(J)<1024 THEN PROCMISSILE (J,3)
220 X=J*320-I : Y=Y(J)+32-P
230 IF X<128 AND X>-40 AND Y>-32 AND Y<2 THEN 280
240 NEXT
250 NEXT
260 CLS : PRINT "MANCATO"
270 END
280 PROCPLANE(I,P,1) : SOUND 0,-15,5,20
290 END
300 DEF PROCPLANE(X,Y,C)
310 GCOL 0,C
320 MOVE X,Y
330 VDU 5,224,225,4
340 ENDPROC
350 DEF PROCMISSILE(N,C)
360 GCOL 0,C
370 MOVE 320*N,32+Y(N)
380 VDU 5,226,4
390 ENDPROC
400 DEF PROCDISPLAY
410 FOR I=1 TO 3
420 MOVE I*320,32
430 VDU 5,227,8,226,4
440 NEXT
450 ENDPROC
```



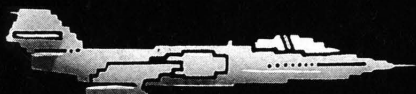
# Missile!: versione Spectrum ZX

```
10 CLS
15 INVERSE 0
20 DIM Y(3)
30 DIM F(3)
40 LET N=1
50 LET MS=8
60 LET PS=INT(RND*6+5)
70 LET P=INT(RND*140+20)
80 GOSUB 400
90 FOR I=PS TO 240 STEP PS
100 LET C=1 : LET X=I-PS
110 GOSUB 300
120 LET C=0 : LET X=I
130 GOSUB 300
140 LET F#=INKEY$
150 IF F#="" OR N>3 THEN GOTO 170
160 LET F(N)=1 : LET N=N+1
170 FOR J=1 TO 3
180 LET C=1 : GOSUB 350
190 IF F(J)=0 OR Y(J)>148 THEN GOTO 240
200 LET Y(J)=Y(J)+MS
210 LET C=0 : GOSUB 350
220 LET X=J*64-I : LET Y=P-Y(J)
230 IF X>-1 AND X<12 AND Y<10 AND Y>-5 THEN GOTO 280
240 NEXT J
250 NEXT I
260 PRINT AT 0,0;"MANCATO"
270 STOP
280 PRINT AT 0,0;"COLPITO"
290 STOP
300 INVERSE C
310 PLOT X,P
320 DRAW 0,8 : DRAW 3,-6
330 DRAW 8,0 : DRAW 2,-2
340 DRAW -13,0 : RETURN
350 INVERSE C
360 PLOT 64*J+4,Y(J)+4
370 DRAW 0,6 : DRAW -2,-2
380 DRAW 2,2 : DRAW 2,-2
390 RETURN
400 FOR A=65 TO 66
410 FOR B=0 TO 7
420 READ C
430 POKE USR CHR$(A)+B,C
440 NEXT B
450 NEXT A
460 FOR J=1 TO 3
470 PRINT AT 21,8*J;"[AB]"
480 NEXT J
490 RETURN
500 DATA 0,0,0,0,255,
255,127,63
510 DATA 0,252,252,252,
255,255,254,252
```



## Missile!: versione VIC 20

```
1 REM*****
2 REM**IL PROGRAMMA E' PER IL VIC20**
3 REM**NON GIRA SUL C64.      *
4 REM*****
10 PRINTCHR$(147)CHR$(5);
20 POKE36879,8
60 DIMY(3),F(3)
70 N=1:MS=2
90 P=INT(RND(1)*9+2)*2
110 FORI=1TO21STEP RND(1)/2+.5
120 GOSUB300
```



```
130 GETF$
140 IFF$=""ORN>3THEN170
150 F(N)=-1
160 N=N+1
170 FORJ=1TO3
180 IFF(J)=0THEN220
190 POKE8164+J*5-Y(J)*22,32
200 IFY(J)=22THEN240
210 Y(J)=Y(J)+1
220 POKE8164+J*5-Y(J)*22,30
230 IFABS(I-J*5)<=1ANDP=Y(J)THEN280
240 NEXT
250 NEXT
260 PRINTCHR$(147);"MANCATO"
270 END
280 PRINTCHR$(147);"COLPITO"
282 POKE36877,0:POKE36878,0
284 FORK=1TO500:NEXT
286 POKE36877,0:POKE36878,0
290 END
300 POKE8163+I-P*22,32
310 POKE8164+I-P*22,121
320 RETURN
```

### Nota speciale per i possessori di Spectrum:

La convenzione usata per mostrare i caratteri grafici in una istruzione PRINT è la seguente:

il carattere [significa: premi il tasto GRAPHICS una sola volta quindi batti i tasti delle lettere seguenti (usando contemporaneamente il tasto SHIFT per lo ZX81) e infine premi di nuovo il tasto GRAPHICS quando incontri].



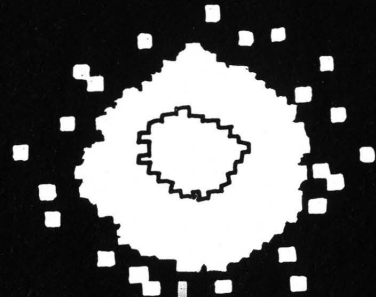
## Missile!: versione Apple

```
10 HOME
20 HGR
30 HCOLOR=3
40 DIM Y(3),F(3)
50 N=1 : MS=5
60 PS=INT(RND(1)*6+4)
70 P=INT(RND(1)*135+11)
80 GOSUB 400
90 FOR I=PS TO 265 STEP PS
100 X=I-PS : Y=159-P : C=0 : GOSUB 300
110 X=I : C=3 : GOSUB 300
120 F$="" : IF PEEK(-16384)>127 THEN GET F$
130 IF F$="" OR N>3 THEN 160
140 F(N)=1
150 N=N+1
160 FOR J=1 TO 3
170 C=0 : GOSUB 350
180 IF F(J)=0 : OR Y(J)>145 THEN 230
190 Y(J)=Y(J)+MS
200 C=3 : GOSUB 350
210 X=J*70-I : Y=P-Y(J)
220 IF X>-1 AND X<15 AND Y>-9 AND Y<5 THEN 270
230 NEXT
240 NEXT
250 VTAB22 : PRINT "MANCATO"
260 END
270 VTAB22 : PRINT "COLPITO"
280 END
300 HCOLOR=C
310 HPLOT X,Y TO X,Y-8
320 HPLOT TO X+3,Y-2 : HPLOT TO X+12,Y-2
330 HPLOT TO X+14,Y : HPLOT TO X,Y
340 RETURN
350 HCOLOR=C
360 HPLOT 70*J,158-Y(J) TO 70*J,154-Y(J)
370 RETURN
400 FOR J=1 TO 3
410 HPLOT 70*J-5,159 TO 70*J+5,159
420 NEXT
430 RETURN
```



# Missile!: versione ZX81

```
10 CLS
20 DIM Y(3)
30 DIM F(3)
40 LET N=1
50 LET P=INT(RND*19+2)*2
60 FOR I=1 TO 48
70 PLOT I,P
80 PLOT I+1,P
90 UNPLOT I-1,P
100 LET B$=INKEY$
110 IF B$="" OR N>3 THEN GOTO 140
120 LET F(N)=1
130 LET N=N+1
140 FOR J=1 TO 3
150 IF F(J)=0 THEN GOTO 190
160 UNPLOT J*16,Y(J)
170 IF Y(J)=42 THEN GOTO 210
180 LET Y(J)=Y(J)+1
190 PLOT J*16,Y(J)
200 IF P=Y(J) AND (J*16=I OR J*16=I+1)
    THEN GOTO 260
210 NEXT J
220 NEXT I
230 CLS
240 PRINT "MANCATO"
250 STOP
260 CLS
270 PRINT "COLPITO"
280 STOP
```





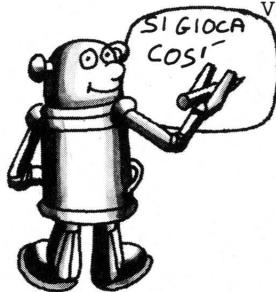
# Aggiunte ai programmi

Eccovi ora alcune idee da usare per eventuali aggiunte ai programmi presentati in questo libro o a quelli scritti da voi stessi. Nella maggioranza dei casi non riuscirete a realizzare queste aggiunte sullo ZX81 con un solo K di memoria, ma dovrete avere invece molto spazio sugli altri computer.

Ricordate che dovrete limitare le vostre aggiunte ai soli numeri di linea ancora liberi nei programmi, oppure rinumerare il programma. Se scegliete quest'ultima soluzione, ricordate di cambiare anche tutte le linee contenenti i GOTO ed i GOSUB.

## Come farvi spiegare i giochi dal computer

Potete aggiungere ad ogni programma una sezione che si occupi di far visualizzare dal computer le istruzioni e le spiegazioni dei giochi. Il modo piú semplice di far ciò, consiste nello aggiungere alcune linee, come quelle sotto riportate, all'inizio del programma e di mettere una subroutine al termine.



```
10 PRINT "TITOLO DEL GIOCO"
11 PRINT "VUOI SAPERE"
12 PRINT "COME SI GIOCA?"
15 INPUT I$
S▲ 17 IF I$(1)="S" THEN GOSUB 1000
VIC 17 IF LEFT$(I$,1)="S"
      THEN GOSUB 1000
```

programma principale

```
1000 PRINT "CIO' CHE DEVI FARE"
1010 PRINT "E' QUESTO..."
1999 RETURN
```

Potete aggiungere istruzioni PRINT a vostro piacimento, ricordandovi però di porre un numero di linea corretto all'inizio di ognuna di esse. Limitate inoltre la lunghezza delle frasi fra virgolette al numero massimo di caratteri consentito dal vostro computer su una sola riga. Non dimenticate di inserire una linea con l'istruzione RETURN al termine del programma altrimenti non funzionerà.

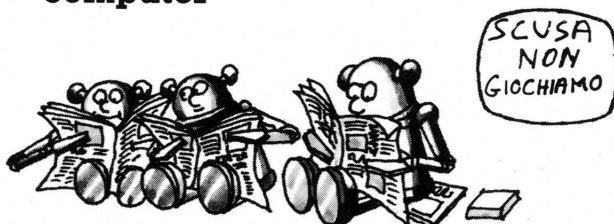
## Come far attendere il computer



Se le vostre istruzioni sono molto lunghe, potrebbe tornarvi comodo inserire questa subroutine che interrompe l'esecuzione di un programma in un punto particolare finché voi non premerete un tasto. In questo modo potete impedire che le istruzioni scorrano via dallo schermo prima che le abbiate potute leggere. Mettete una linea contenente un GOSUB al punto in cui volete che il programma si fermi ed inserite questa subroutine alla fine.

```
1000 PRINT "PREMI UN TASTO PER
CONTINUARE";
S▲ 1010 IF INKEY#="" THEN GOTO 1010
* 1010 I$=GET$
● 1010 GET I$
VIC 1010 GET I$ : IF I$="" THEN GOTO 1010
1020 PRINT
1030 RETURN
```

## Come farvi parlare dal computer



Potete fare in modo che il computer vi faccia delle domande e reagisca alle vostre risposte. Ad esempio, ecco una aggiunta che impedirà al computer di giocare se il vostro nome non inizia con la lettera "J".

```
1 PRINT "COME TI CHIAMI?"
2 INPUT I$
S▲ 3 IF I$(1) <> "J" THEN GOTO 1000
*●VIC 3 IF LEFT$(I$,1) <> "J" THEN GOTO 1000
4 PRINT "OK-VUOI GIOCARE"
5 PRINT "SEI PRONTO?"
6 INPUT J$
S▲ 7 IF J$(1) <> "S" THEN GOTO 5
*●VIC 7 IF LEFT$(I$,1) <> "S" THEN GOTO 5
```

programma principale

```
1000 PRINT "QUESTO GIOCO"
1010 PRINT "E' SOLO PER"
1020 PRINT "CHI HA NOME INIZIANTE"
1030 PRINT "CON J"
```



Eccone ora un'altra con cui il computer vi chiederà se siete abbastanza coraggiosi da affrontare il gioco.

```

10 PRINT "GIOCO TERRIFICANTE"
12 PRINT "HAI IL CORAGGIO"
14 PRINT "DI AFFRONTARE"
15 PRINT "IL MOSTRO VERDE?"
16 INPUT I$
S▲17 IF I$(1)="S" THEN GOTO 20
*■●VIC17 IF LEFT$(I$,1)="S" THEN GOTO 20
18 PRINT "CODARDO"
19 STOP

```

Potete combinare questa aggiunta con la subroutine delle istruzioni, prendendo le linee dalla 11 alla 17 di quella subroutine e mettendole alle linee 20-26 di quest'ultimo programma. Potete quindi iniziare il programma principale alla linea 30 e poi aggiungere la subroutine di istruzioni alla fine.

## Volete giocare ancora?

Invece di battere RUN ogni volta che giocate, potete fare in modo che il computer vi chieda se volete giocare di nuovo. Mettete queste linee al termine del programma, prima della istruzione STOP.

```

1000 PRINT "VUOI GIOCARE ANCORA?"
1010 INPUT I$
S▲1020 IF I$(1)="S" THEN RUN
*■●VIC1020 IF LEFT$(I$,1)="S" THEN RUN
1030 PRINT "OK-CIAO"
1040 STOP

```

**Cambiate i numeri delle linee a seconda del vostro programma.**



## Aggiunta di effetti sonori

I computer BBC, VIC 20, ZX Spectrum ed alcuni Apple sono in grado di produrre suoni e voi potete aggiungere linee ai vostri programmi per ottenere effetti in punti appropriati. Potete aggiungere un'esplosione, ad esempio, o un motivetto da suonare in caso di vittoria. Ogni computer ha istruzioni particolari per produrre i suoni, quindi dovrete basarvi sul vostro manuale. In alcuni casi basta

aggiungere una sola linea al programma nel punto in cui si vuole il sonoro; in altri servono molte linee ed è bene allora inserirle come subroutine.

Come esempio, eccovi il suono di uno sparo per il BBC. Potete cercare il punto migliore per inserirlo in un programma, ma ricordate di assegnare un numero di linea per farlo funzionare:

```
SOUND 0, -15, 5, 10
```

Al termine del manuale del VIC troverete alcune utili subroutines per effetti tipo "raggio laser", "esplosione" e "allarme rosso". Mettete una linea con un GOSUB nel punto in cui volete che compaia il suono, numerate la subroutine ed aggiungetevi al termine un RETURN.



## Nota speciale per i proprietari di BBC e Spectrum



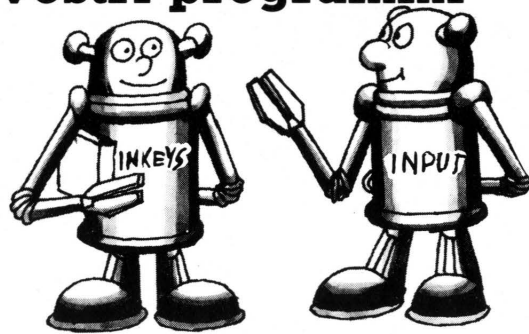
Se avete un computer BBC oppure uno Spectrum, potrete trovare alcuni dei giochi di questo libro troppo veloci per voi. Vicino alle descrizioni di questi giochi troverete dei riquadri in cui sono contenute le istruzioni necessarie per cambiare la velocità di esecuzione. Ricordate: per rallentare il gioco dovete sempre usare un numero più alto. I modelli più recenti della BBC possono raggiungere velocità di esecuzione anche doppie rispetto ai modelli più vecchi e questo potrebbe far apparire impossibili a prima vista i giochi. Preparatevi a fare cambiamenti anche notevoli alla velocità per rimediare!



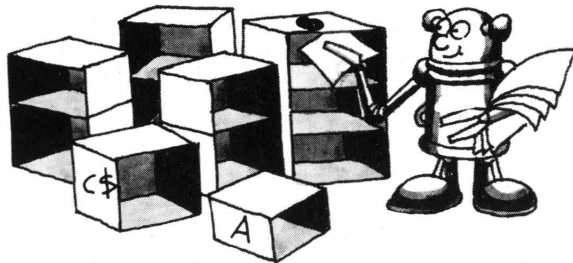
# Scrivete voi stessi i vostri programmi

Usando i giochi di questo libro, vi capiterà di voler apportare sempre nuove modifiche o addirittura di voler scrivere giochi nuovi. Nelle pagine seguenti troverete dei suggerimenti su come intraprendere ciò.

Prima di iniziare è consigliabile fermarsi un momento a pensare a ciò che il vostro computer può fare e a ciò che non può fare.



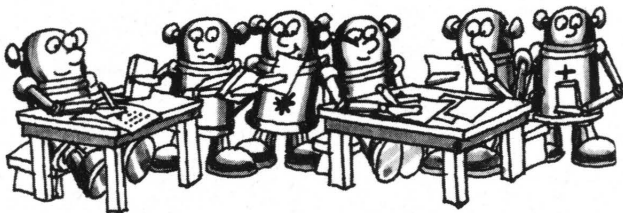
\* Può chiedervi informazioni.



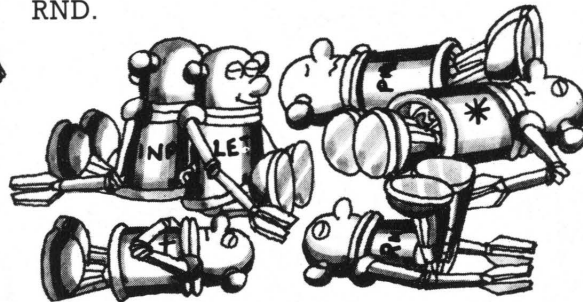
\* Può memorizzare informazioni.



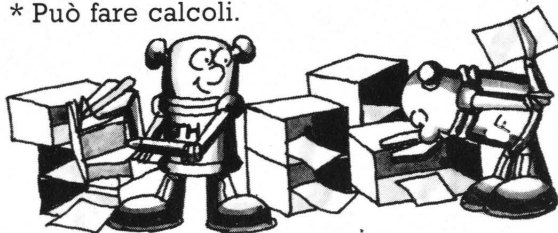
\* Può scegliere numeri a caso usando RND.



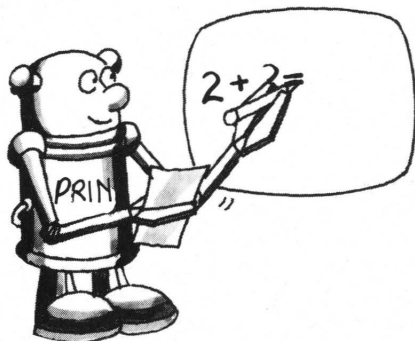
\* Può fare calcoli.



\* Non può fare nulla che non gli diciate.



\* Puoi prendere decisioni confrontando informazioni in vari modi.



\* Può fare esattamente ciò che gli ordinate usando correttamente il suo linguaggio, anche se gli ordinate qualcosa di sciocco.

\* Può dirvi i risultati dei suoi calcoli e ciò che è immagazzinato in memoria.

Ricordate, quando state cercando di creare un gioco, di non includere azioni che il vostro computer non è in grado di eseguire.



## Progettare un gioco

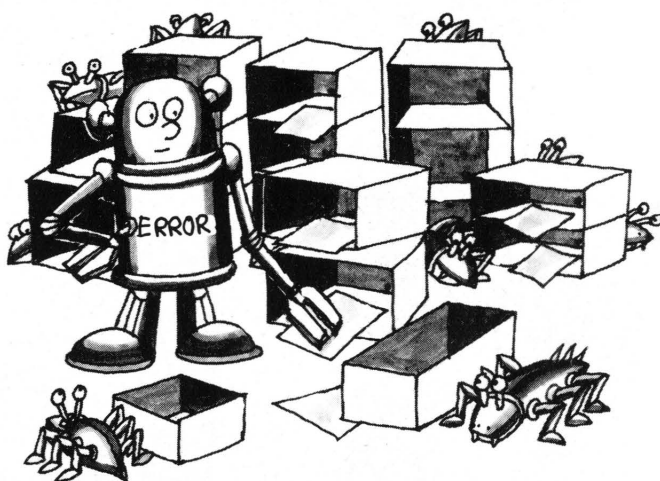
Prima di istruire il computer su come far funzionare il gioco dovete sapere voi come si gioca e conoscere esattamente le regole. Il computer ha bisogno di una serie di semplici istruzioni logiche, quindi cercate di avere bene in mente il gioco e di descriverlo su un foglio e quindi suddividetelo in passi semplici. Poi scrivete un piano (in italiano — non cercate di usare il BASIC a questo punto) di tutti gli stadi del gioco nella loro successione. Qui di seguito vi è un piano per un semplice gioco di tiro, del tipo di quelli di cannonate contro le navi pirata o di raggi laser contro gli invasori alieni, per darvi un'idea.

### PIANO

- 1) STAMPA TITOLO ED ISTRUZIONI
- 2) SCEGLI UN OBIETTIVO PER QUESTO GIOCO.
- 3) INIZIA UN CICLO PER DARE N TENTATIVI AL GIOCATORE
- 4) RICEVI UNO SPARO DAL GIOCATORE
- 5) CONTROLLA SE LO SPARO È ANDATO A SEGNO
- 6) STAMPA UN MESSAGGIO SECONDO LA PRECISIONE DEL TIRO
- 7) PASSA AL TENTATIVO SEGUENTE SE IL PRECEDENTE È FALLITO

## Come scrivere il programma

Lo stadio successivo è la conversione del vostro piano in BASIC. Ogni passo del vostro piano può usare parecchie linee in BASIC. Non dimenticate di lasciare numeri mancanti durante la numerazione delle vostre linee di programma in modo da poter tornare indietro e inserire altre linee se necessario. Fate una prima stesura del programma su carta e quindi provatelo sul computer. Il vostro computer troverà gli errori molto più velocemente di voi e potrà suggerirvi cosa non va. Ricordate che correggere i programmi è un lavoro lungo e noioso anche per programmatori esperti, quindi non aspettatevi di riuscire al primo



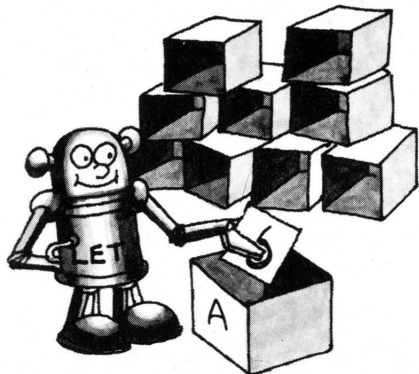
tentativo a scrivere un programma senza errori.

Una volta che il grosso del programma funziona, potete farvi delle aggiunte. Punteggi, commenti extra, altri bersagli, ecc. possono essere aggiunti in un secondo tempo. Potete aggiungere parti dei programmi di questo libro ai vostri giochi. Non aspettatevi di scrivere giochi originali ed eccitanti di punto in bianco. Seguite idee semplici e siate pronti a riadattarle man mano che procedete. Troverete a volte che avete incluso nei programmi cose facili per un uomo, ma impossibili per un computer. Diventando sempre più esperti vi verrà istintivo scrivere programmi contenenti unicamente azioni che il vostro computer può eseguire.



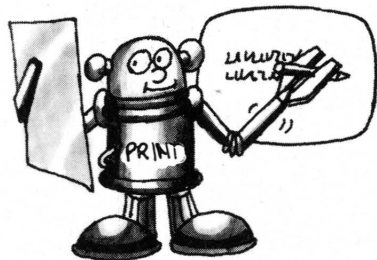
# Riassunto del BASIC

Questa sezione contiene un elenco di alcuni vocaboli BASIC comunemente usati e descrive cosa essi comandano al computer di fare e come vengono usati. La maggior parte di essi sono stati usati nei programmi di questo libro, potete così controllare voi stessi come essi vengono impiegati nei giochi. Non tutti i vocaboli possono venir usati su tutti i computer menzionati in questo libro, la carta di conversione a pag. 46 mostra le alternative da usare.



**LET** dice al computer di etichettare una sezione della sua memoria e di inserirvi un valore particolare. Ad es. **LET A=6** significa: etichetta con "A" una sezione di memoria e inseriscivi il valore 6. "A" è detta una "variabile" e mettervi qualcosa dentro si dice "assegnare un valore ad una variabile".

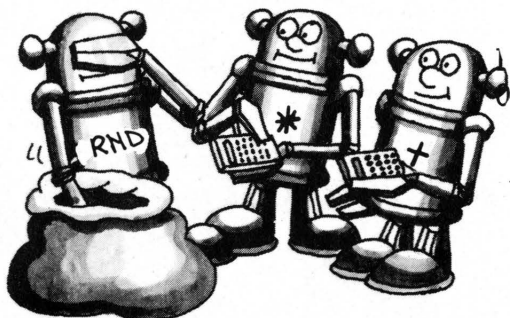
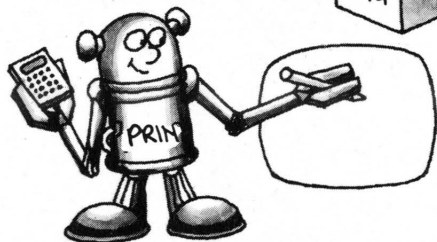
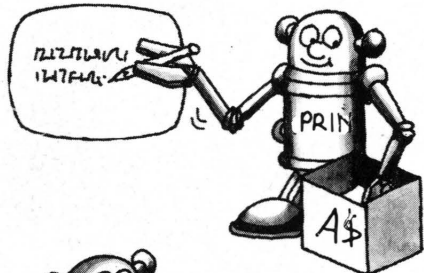
Alcune etichette di variabili sono seguite dal segno del dollaro, ad es. **A\$**. Questo significa che esse rappresentano delle "stringhe", che possono contenere un qualsiasi numero di caratteri, compresi lettere, numeri e simboli.



**PRINT** dice al computer di mostrare sullo schermo degli oggetti e potete usarlo in parecchi modi: un messaggio fra virgolette che segua un'istruzione **PRINT** verrà visualizzato esattamente come lo avete battuto. La parte fra virgolette non deve essere in BASIC, può essere qualsiasi cosa volete.

**PRINT** seguito da un'etichetta di variabile, ad es. **PRINT A** o **PRINT A\$** dice al computer di visualizzare sullo schermo il contenuto di quella variabile.

**PRINT** può anche fare eseguire dei calcoli, visualizzandone il risultato, ad es. **PRINT 6\*4** farà scrivere 24 al computer. Potete usare **PRINT** da solo per lasciare una linea di spazio.



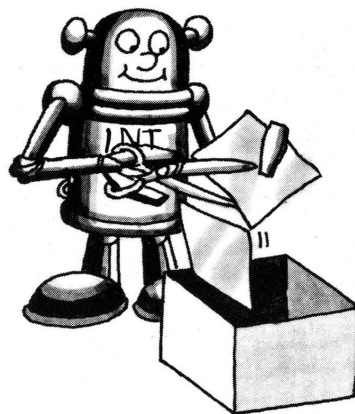
**RND** dice al computer di scegliere un numero a caso. A seconda del computer ci sono diverse forme di **RND** che potete vedere nella tabella di conversione a pag. 46. Sui computer Sinclair, **RND** da solo produce un numero compreso fra 0 e 0,99999999. Potete variare i limiti del numero scelto per mezzo di una moltiplicazione di un'addizione. Ad es. **RND\*20** produce un numero fra 0 e 19,99999999, mentre **RND\*20 + 1** produce un numero fra 1 e 20,99999999.

Vedere l'istruzione **INT** per sapere come produrre solo numeri interi. Vedere l'istruzione **CHR\$** per sapere come produrre lettere ed altri simboli a caso.

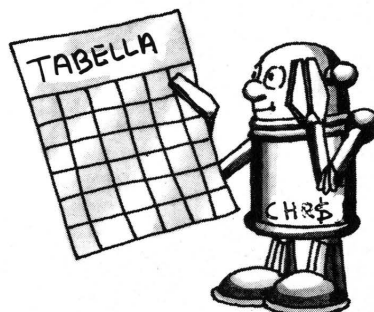


**INT** è un'abbreviazione di intero, che sta per numero intero. Se ci si riferisce a numeri positivi, dice al computer di trascurare tutto ciò che si trova alla destra della virgola, cioè del punto decimale. Ad es.  $\text{INT}(20,999)$  è uguale a 20. Per numeri negativi viene ignorato tutto ciò che è alla destra del punto e aumentato di uno il numero alla sinistra, ad es.  $\text{INT}(-3.6)$  è uguale a  $-4$ .

**INT** è spesso usato insieme a **RND** in questo modo:  $\text{INT}(\text{RND} * 20 + 1)$ , che dice al computer di scegliere un numero intero compreso fra 1 e 20.

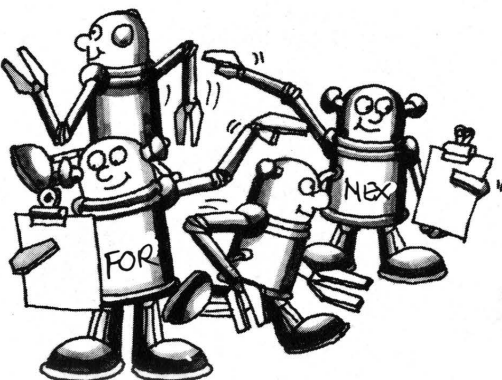


**CHR\$** converte i numeri in lettere. Ad eccezione dello ZX81, tutti i computers in questo libro usano l'insieme di caratteri da tastiera ASCII\*, in cui a ciascun carattere corrisponde un certo numero. Ad es. la lettera A ha numero di codice 65 e l'istruzione  $\text{PRINT CHR}\$(65)$  farà visualizzare una A sullo schermo. Potete usare **CHR\$** insieme a **INT** e **RND** per far scegliere al computer lettere casuali in questo modo:  $\text{CHR}\$(\text{INT}(\text{RND} * 26 + 65))$ . Questa linea produrrà lettere casuali su uno ZX81 Spectrum (vedete la tabella di conversione per gli altri computer).



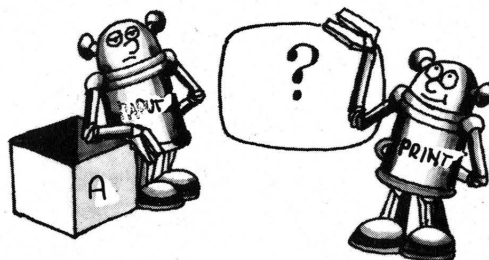
**FOR** è usato per iniziare un ciclo che faccia ripetere al computer una certa parte di programma un certo numero di volte. Deve essere seguito da una variabile (come ad es. V per il numero di volte che un giocatore può tentare in un gioco) e la variabile deve avere dei valori iniziali e finali (come ad es. 1 e 10).

Il termine del ciclo è segnalato da una linea contenente **NEXT** (**NEXT G** nel nostro esempio), che incrementa ogni volta di 1 il valore della variabile e quindi rinvia il computer alla linea contenente **FOR**. Quando la variabile assume il suo valore finale il computer ignora la linea **NEXT** e passa a quella che la segue nel programma. Ogni **FOR** deve avere un **NEXT** altrimenti avrete un messaggio di errore dal computer.



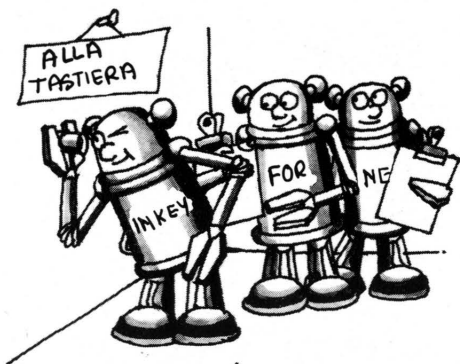
**INPUT** etichetta uno spazio nella memoria del computer, fa stampare un punto di domanda sullo schermo ed attende che voi battiate qualcosa da inserire in questo spazio di memoria. Il programma non proseguirà finché non avrete premuto **RETURN**, **ENTER** o **NEWLINE**.

Con **INPUT** potete usare variabili numeriche o stringhe ma se usate una variabile numerica il computer non accetterà lettere!



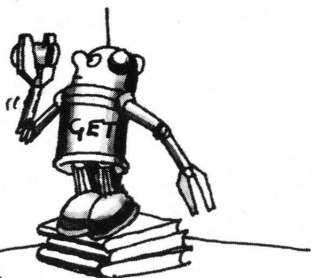
\* Vedi tabella pag. 45





**INKEY\$** controlla la tastiera per vedere se è premuto un tasto e quale. Non attende che voi battiate, come INPUT, ed è solitamente usato in cicli di controllo della tastiera. Questo è dovuto al fatto che, a causa della velocità del computer, non avreste la possibilità di premere un tasto nel tempo che il computer impiega ad eseguire un controllo.

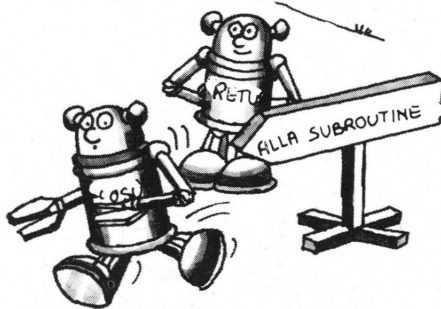
Se non avete premuto alcun tasto prima della fine del ciclo, il computer rileverà una stringa che non contiene nulla (detta una stringa "vuota" o "nulla"). NB: Apple e VIC non usano INKEY\$.



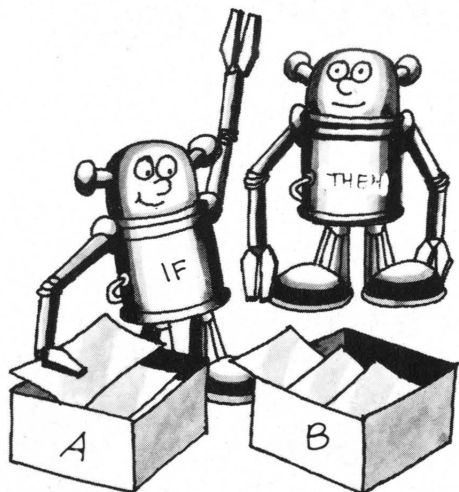
**GET** è usato dai computer VIC e Pet invece di INKEY\$.



**GOTO** fa in modo che il computer salti, su o giù per il programma ignorando le linee intermedie. Dovete scrivere il numero di linea a cui volete che il computer passi dopo l'istruzione GOTO.



**GOSUB** dice al computer di lasciare il programma principale e passare ad una subroutine. GOSUB deve essere seguito dal numero della prima linea della subroutine. Al termine della subroutine dovete avere una linea con RETURN. Questa manda il computer alla linea immediatamente successiva a quella con GOSUB. Un GOSUB senza un RETURN in un programma produrrà un messaggio d'errore da parte del computer.



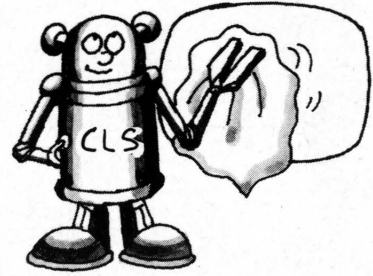
**IF ... THEN** dice al computer di decidere se un'espressione è vera o falsa e di fare cose differenti a seconda della risposta. Si usa con i seguenti simboli, oltre che con AND e OR:

- = uguale a
- < minore di
- > maggiore di
- <= minore o uguale a
- >= maggiore o uguale a
- <> diverso da

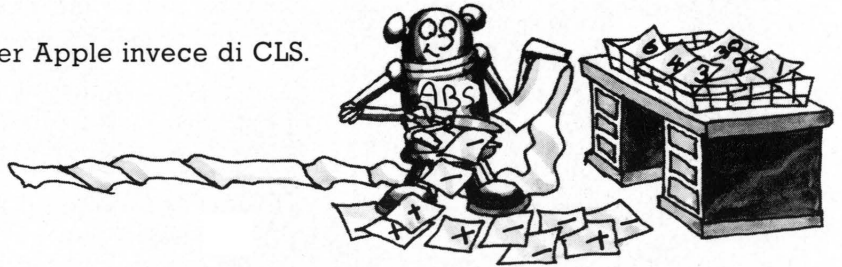
Se il computer stabilisce che un'espressione è vera, esegue poi l'istruzione che segue THEN. Se decide che è falsa, ignora il resto della linea e passa alla successiva.



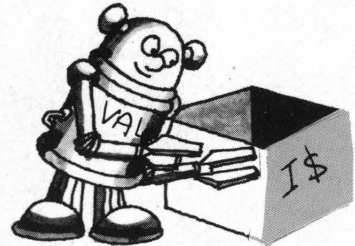
**CLS** è usato per cancellare tutto lo schermo senza rimuovere o cambiare nulla nella memoria. È utile per rimuovere il listing dallo schermo all'inizio di un'esecuzione, o in giochi in cui il giocatore deve reagire a qualcosa visto per un tempo limitato. (NB: Apple e VIC non usano CLS-vedere la tabella di conversione).



**HOME** è usato dai computer Apple invece di CLS.

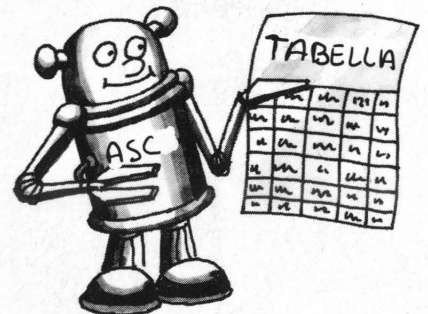


**ABS** ignora i segni + e - davanti ai numeri e prende i loro valori assoluti. Ad es. ABS (- 10) è 10 ed ABS (+ 10) è ancora 10.



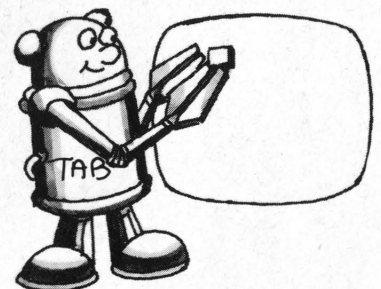
**VAL** prende il valore numerico di numeri scritti sotto forma di stringhe. In realtà dice al computer di ignorare il segno del dollaro e di trattare la stringa come un'ordinaria variabile numerica. Ad es. se I\$ = "60" allora VAL(I\$) è il numero 60.

**ASC** converte un carattere nel suo numero di codice ASCII corrispondente, ad es. ASC("3") dà 51. L'espressione fra parentesi deve essere una stringa, ad es. ASC(A\$) oppure ASC("20"). NB: ZX81 e ZX Spectrum non usano ASC sebbene lo Spectrum usi il codice ASCII.

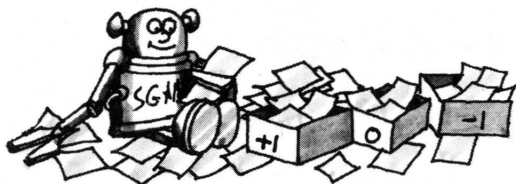


**CODE** è usato dallo ZX81 e dallo Spectrum al posto di ASC. Come ASC deve sempre essere seguito da una stringa. Ricordate che lo ZX81 usa numeri di codice differenti da quelli degli altri computers.

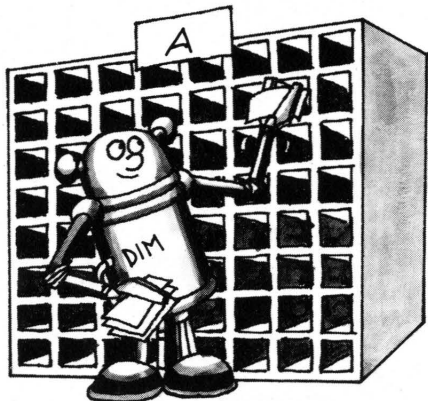
**TAB** muove il cursore attraverso lo schermo fino a una colonna specificata. È solitamente usato con PRINT per visualizzare qualcosa al centro dello schermo. Il numero di spazi di cui volete muovere il cursore è posto fra parentesi dopo TAB. Il massimo numero che potete usare dipende dalla larghezza di schermo del vostro computer.







**SGN** dice al computer di scoprire il segno di un numero. Produce  $-1$  per i numeri negativi e  $0$  per lo zero e  $+1$  per i numeri positivi. Ad es.  $SGN(-30)$  è uguale a  $-1$ ,  $SGN(7)$  è uguale a  $+1$  e  $SGN(0)$  è  $0$ .

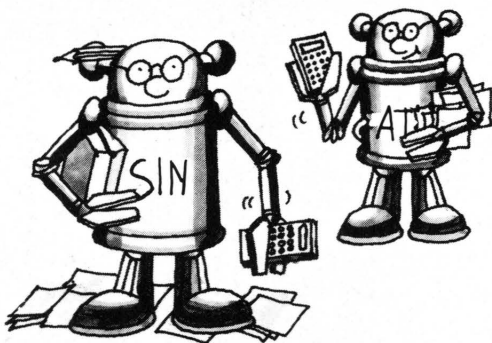


**DIM** dice al computer quanto spazio di memoria sarà necessario per una "matrice" (una riga o una griglia). Ad es.  $DIM X(6)$  dice al computer di riservare un'area abbastanza grande da contenere una riga di 6 elementi etichettata X.  $DIM A(8,8)$  richiede uno spazio di memoria etichettato A e grande abbastanza da contenere 8 elementi in orizzontale ed 8 in verticale. Il numero di elementi di dati usati nel programma deve corrispondere ai numeri fra parentesi che seguono DIM o si avrà un messaggio di errore.

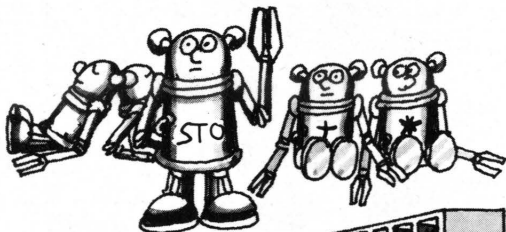


**SQR** prende le radici quadrate dei numeri. Ad ad.  $SQR(16)$  dà 4.

**SIN** calcola il seno di un angolo. In un triangolo rettangolo la lunghezza del lato opposto ad un angolo diviso per la lunghezza dell'ipotenusa (il lato opposto all'angolo retto) è il seno di quell'angolo. Quando usate SIN in un programma, l'angolo dovrà essere misurato in radianti, non in gradi.



**ATN** è una delle funzioni trigonometriche che i computers possono calcolare (v. sopra anche SIN). Sta per arcotangente ed è importante ricordare che dà risultati in radianti e non in gradi. Dovrete usare un testo di matematica per scoprire come funziona, se non lo sapete già.



**STOP** dice al computer di non procedere oltre. Computers diversi dallo ZX81 possono usare END al suo posto.

**PEEK** è un modo per trovare cosa c'è in un'area specifica della memoria del computer. Dovete usarlo con un numero che specifica un indirizzo della memoria. NB: non usato dai BBC.



**POKE** è un modo speciale per inserire informazioni nella memoria nel computer usando un indirizzo di memoria. NB: non usato dai BBC.



## Tabella ASCII

Codice	Caratteri ASCII	Codice	Caratteri ASCII
32	space	62	>
33	!	63	?
34	"	64	@
35	#	65	A
36	\$	66	B
37	%	67	C
38	&	68	D
39	'	69	E
40	(	70	F
41	)	71	G
42	*	72	H
43	+	73	I
44	,	74	J
45	-	75	K
46	.	76	L
47	/	77	M
48	0	78	N
49	1	79	O
50	2	80	P
51	3	81	Q
52	4	82	R
53	5	83	S
54	6	84	T
55	7	85	U
56	8	86	V
57	9	87	W
58	:	88	X
59	;	89	Y
60	<	90	Z
61	=		

## Codice tabella ZX81

Codice	Caratteri ZX81	Codice	Caratteri ZX81
11	"	41	D
12	£	42	E
13	\$	43	F
14	:	44	G
15	?	45	H
16	(	46	I
17	)	47	J
18	>	48	K
19	<	49	L
20	=	50	M
21	+	51	N
22	-	52	O
23	*	53	P
24	/	54	Q
25	;	55	R
26	,	56	S
27	.	57	T
28	0	58	U
29	1	59	V
30	2	60	W
31	3	61	X
32	4	62	Y
33	5	63	Z
34	6		
35	7		
36	8		
37	9		
38	A		
39	B		
40	C		

## Tabella dimensioni schermi

	Massimo numero di righe	Massimo numero di colonne
VIC 20	22	23
TRS-80	64	16
BBC	20/40/80	16/24/32
ZX81	32	22
ZX Spectrum	32	22
Apple	40	25

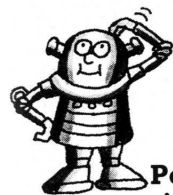


## Tabella di conversione

Questa tabella di riferimento rapido mostra alcune delle differenze fra i BASIC usati dalle macchine in questo libro. Non comprende le istruzioni per grafica, suono e colore dal momento che variano enormemente da macchina a macchina. Notate anche che sebbene la maggior parte dei computers (escluso il BBC) usino PEEK e POKE, essi non usano lo stesso sistema di indirizzi della memoria, così i numeri usati con PEEK e POKE devono essere modificati per ogni computer.

	BBC	VIC	Apple	TRS-80	ZX Spectrum	ZX81
Scegli un numero casuale fra 0 e 0,99999999	RND(1)	RND(1)	RND(1)	RND(0)	RND	RND
Scegli un numero casuale fra 1 e N	RND(N)	RND(1)*N+1	RND(1)*N+1	RND(N)	RND*N+1	RND*N+1
Scegli una lettera a caso fra A e Z	CHR\$(RND(26)+64)	CHR\$(INT(RND(1)*26+65))	CHR\$(INT(RND(1)*26+65))	CHR\$(RND(26)+64)	CHR\$(INT(RND*26+65))	CHR\$(INT(RND*26+38))
Cancello lo schermo	CLS	PRINT CHR\$(147).	HOME	CLS	CLS	CLS
Controlla se è premuto un tasto qualsiasi	INKEY\$(N)	GET X\$	X\$ = " " IF PEEK(-16384) > 127 THEN GET X\$	INKEY\$	INKEY\$	INKEY\$
Converti i caratteri in numeri di codice	ASC("X") (usando codice ASCII)	ASC("X") (usando codice ASCII)	ASC("X") (usando codice ASCII)	ASC("X") (usando codice ASCII)	CODE("X") (usando codice ASCII)	CODE("X") (usando codice ZX81)
Muovi il cursore in alto	PRINT CHR\$(11)	PRINT CHR\$(145)	CALL - 998	PRINT CHR\$(27)	PRINT CHR\$(11)	PRINT CHR\$(112)
Muovi il cursore in basso	PRINT CHR\$(10)	PRINT CHR\$(17)	PRINT CHR\$(10)	PRINT CHR\$(26)	PRINT CHR\$(10)	PRINT CHR\$(113)
Muovi il cursore a sinistra	PRINT CHR\$(8)	PRINT CHR\$(157)	PRINT CHR\$(8)	PRINT CHR\$(24)	PRINT CHR\$(8)	PRINT CHR\$(114)
Muovi il cursore a destra	PRINT CHR\$(9)	PRINT CHR\$(29)	PRINT CHR\$(21)	PRINT CHR\$(25)	PRINT CHR\$(9)	PRINT CHR\$(115)
Prendi i primi N caratteri di una stringa	LEFT\$(A\$,N)	LEFT\$(A\$,N)	LEFT\$(A\$,N)	LEFT\$(A\$,N)	A\$(1 TO N)	A\$(1 TO N)
Prendi gli ultimi N caratteri di una stringa	RIGHT\$(A\$,N)	RIGHT\$(A\$,N)	RIGHT\$(A\$,N)	RIGHT\$(A\$,N)	A\$(N TO )	A\$(N TO )
Prendi gli N caratteri centrali di una stringa	MID\$(A\$,N1,N2)	MID\$(A\$,N1,N2)	MID\$(A\$,N1,N2)	MID\$(A\$,N1,N2)	A\$(N1 TO N2)	A\$(N1 TO N2)





# Risposte

Potrete trovare alcune delle vostre risposte ai rompicapo diverse da quelle date qui. Se esse funzionano sul vostro computer questo non ha importanza, ma controllate lo stesso se sono semplici e pulite come le risposte del libro.

## Pagina 5: Missile Robot

La linea 90 dice al computer quanti cicli deve eseguire per ottenere la vostra risposta. Quindi se volete piú possibilitá di indovinare la lettera segreta, cambiate l'ultimo numero nella linea 90 con uno maggiore, se ne volete meno con uno minore.

## Pagina 7: Messaggio vitale

In questo programma le linee 150 e 160 sono un ciclo di ritardo; lasciano il computer inattivo per un certo lasso di tempo prima di passare all'istruzione successiva, che cancella il messaggio dallo schermo. Per lasciare il messaggio piú a lungo, dovete lasciare il computer inattivo piú a lungo: si può ottenere ciò cambiando l'ultimo numero della linea 150 con uno maggiore.

## Pagina 9: Duello

Per fare in modo che anche il computer possa sbagliare, aggiungete le linee seguenti a quelle del riquadro "Come complicare il gioco"

```
155 IF RND<.1 THEN GOTO 250
250 PRINT "HA SBAGLIATO"
260 GOTO 90
```

La linea 155 dovrà forse essere cambiata usando RND(1) o RND(0) a seconda del vostro computer.

## Pagina 11: Battaglia nel deserto

Aggiungete le linee seguenti al programma per aumentare la possibilitá che i robot vi vedano:

```
175 IF RND<.05 THEN GOTO 250
250 PRINT "TI HANNO VISTO-RAGGGHHHH"
260 STOP
```

Cambiate RND nella linea 175 nella versione adatta al vostro computer. Potete cambiare .05 con qualsiasi numero fra 0 e 0.999 ma ricordate che all'aumentare del numero crescono le probabilitá che i robot vi vedano.

## Pagina 13: Battaglia al castello del traditore

Questo rompicapo è difficile, quindi non preoccupatevi se non siete riusciti a risolverlo. Provate comunque la soluzione: aggiungete queste linee per prendere i due bersagli.

```
55 LET P=INT(RND+.5)*4+1
S▲ 57 LET P#=CHR$(P+51)
■●VIC 57 LET P#=CHR$(P+78)
70 IF L=T THEN LET R#=R#+P#
180 LET S=S+P
```

## Pagina 15: I robot invasori

Potete avere 100 punti per U, V, W, X o Y e 10 per gli altri con queste modifiche

```
220 LET H=H+10
225 IF P#>"T" AND P#<"Z" THEN LET H=H+90
```

## Pagina 17: Arma segreta

Potete aggiungere il sistema di punteggio del rompicapo inserendo queste linee in piú nel programma.

```
15 LET S=0
145 IF Z<=3 THEN LET S=S+1
190 GOTO 217
215 LET S=S+10
217 PRINT "TOTALE ";S;" PUNTI"
```



## **Pagina 19:**

### **Fuga**

Questo è il modo più semplice di aggiungere tre robot al gioco, sebbene divenga piuttosto difficile vincere.

```
22 FOR R=1 TO 3
25 PRINT "ROBOT " ;R
300 NEXT R
310 STOP
```

## **Pagina 21:**

### **Caccia al pirata**

Per rendere le probabilità del pirata di colpirvi uguali alle vostre di colpire lui cambiate lo 0.7 della linea 290 in 0.5.

## **Pagina 23:**

### **Bombardiere supersonico**

Per colpire più bersagli contemporaneamente, cambiate il 5 nelle linee 30, 70 e 120 con un numero maggiore. Dovete però mettere lo stesso numero in tutte e tre le linee.

## **Pagina 25:**

### **Iceberg**

Potete ingrandire la griglia cambiando tutti gli 8 nelle linee 20, 50, 70, 80, 110, 120, 160, 170, 290 e 310 con un numero maggiore (usate lo stesso numero ogni volta). Per avere più iceberg, cambiate il 4 nella linea 30 con un numero maggiore.

© Copyright per l'edizione originale Usborne Publishing Ltd — 1982  
© Copyright per l'edizione italiana Gruppo Editoriale Jackson — 1985  
Supervisione Tecnica: Daria Gianni  
Impaginazione: Cristina De Venezia  
Fotocomposizione: Composit - Pisa  
Stampa: Grafika 78

Il nome Usborne e il marchio  sono marchi registrati dalla  
Usborne Publishing Ltd., 20 Garrick Street, London WC2E 9BJ, England.

Tutti i diritti sono riservati. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.







# Speedy Computer

Il calcolatore è anche (o soprattutto?) una macchina divertente. Si può giocare con lui, gli si possono fare delle domande, lo si può usare per scrivere una poesia o per suonare.

Questa colorata serie di libri vi propone alcune delle cose più eccitanti che si possono fare con un calcolatore e vi spiega come farle.

Scritti in linguaggio chiaro e comprensibile a chiunque, arricchiti da una moltitudine di illustrazioni, questi libri rappresentano una spiritosa introduzione al mondo dei computer per chi comincia da zero.

## **Battaglie con il computer**

Qui il pacifismo non serve. Si gioca alla guerra per imparare il BASIC, quindi ben venga.

## altri volumi di questa collana

### **LA RIVOLUZIONE INFORMATICA**

Come i computer e le nuove tecnologie hanno rivoluzionato il mondo della comunicazione.

### **I ROBOT**

Una parola, sconosciuta qualche decina di anni fa, che oggi rappresenta una realtà nelle fabbriche.

### **ENTRIAMO NEL CHIP**

Anche un piccolo "chip" può portare una rivoluzione. La sua storia è affascinante come un racconto.